

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»
ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ФАКУЛЬТЕТ ЕКОНОМІКИ І МЕНЕДЖМЕНТУ
ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ

Допускається до захисту
Завідувач кафедри _____ О.О. Ємець
(підпис)

«_____» _____ 2020 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО БАКАЛАВСЬКОЇ РОБОТИ
на тему
РОЗРОБКА САЙТУ «Broker»

зі спеціальності 122 «Комп'ютерні науки та інформаційні технології»

Виконавець роботи Тищенко Владислав Юрійович

_____ «__» _____ 2020 р.

Науковий керівник к.ф.-м.н., доц., Чілікіна Тетяна Василівна

_____ «__» _____ 2020 р.

ПОЛТАВА 2020

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І	
ТЕРМІНІВ	3
ВСТУП.....	4
1. ПОСТАНОВКА ЗАДАЧІ.....	6
2. ІНФОРМАЦІЙНА ЧАСТИНА.....	8
2.1. Історія розвитку веб-сайтів.....	8
2.2. Основні етапи створення веб-сайтів.....	10
2.3. Сучасні веб-технології.....	12
2.4. Етапи розробки веб-сайта.....	16
2.4.1. Вибір програмних засобів для розробки веб-сайта.....	16
2.4.2. Обґрунтування вибору системи управління вмістом.....	18
3. ТЕОРЕТИЧНА ЧАСТИНА.....	21
3.1. Розробка веб-дизайну сайту.....	21
3.2. Структура сайту	23
3.3. Контент та його наповнення.....	25
4. ПРАКТИЧНА ЧАСТИНА.....	28
4.1. Встановлення на сервер CMS WordPress	28
4.2. Встановлення платного шаблону сайту.....	31
4.3. Встановлення необхідних плагінів на сайт	33
4.4. Розробка функціоналу сайту	35
4.5. Тестування сайту.....	43
ВИСНОВКИ.....	45
СПИСОК ЛІТЕРАТУРИ.....	46
ДОДАТОК А. Програма.....	48

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Web	система доступу до пов'язаних між собою документів на різних комп'ютерах, підключених до Інтернету
HTML	HyperText Markup Language – Мова розмітки гіпертекстових документів.
CSS	Cascading Style Sheets (Каскадні таблиці стилів).
CMS	Content Management System (система керування вмістом).
БД	база даних.
FTP	File Transfer Protocol (протокол передачі файлів).
WYSIWYG	What You See Is What You Get (що бачиш, те й отримуєш).
PHP	Hypertext Preprocessor (гіпертекстовий препроцесор).
SEO	search engine optimization (пошукова оптимізація сайту).
XML	eXtensible Markup Language (розширювана мова розмітки).
WP	WordPress (система керування вмістом з відкритим кодом, яка застосовується для створення веб-сайтів).
Back-end	це серверна частина сайту, загрузка інформації, та її опрацювання.
Front-end	це інтерфейс для взаємодії між користувачем і back end.
JS	JavaScript – скриптова мова програмування.
AJAX	Asynchronous JavaScript & XML – підхід до побудови користувацьких інтерфейсів веб-застосунків.

ВСТУП

Самий перший сайт в світі був створений в 1991 році. Автором сайту був Тім Бернерс-Лі. Найперша інформація, що була опублікована на цьому сайті було опис нової технології World Wide Web. Тоді світ в перше побачив прототип нині популярного мови розмітки HTML.

У наш час сайт може бути візитною карткою підприємства, фірми, вашим портфоліо. Також існують Інтернет-магазини, файлообмінники, соціальні мережі та онлайн сервіси, прикладом яких є наше дистанційне навчання [1-4].

За допомогою сайту можна швидко коригувати та поширювати інформацію, таким образом люди, які відвідали ваш сайт будуть володіти найостаннішими даними. За допомогою сайту, організація має можливість донести величезну кількість інформації до людей. Жоден інший мас-медіа не здатний дати такого приголомшуючого ефекту.

Сучасні сайти можна порівняти з вже забутими газетами або журналами, головною відмінністю яких є швидкий та безкоштовний доступ до актуальної інформації в будь-якій точці світу, навіть з мобільних пристроїв.

Найважливіше завдання сайту – донести інформацію до користувачів і потенційних клієнтів. Останнім часом, все більше і більше людей користуються інтернетом, читають новини, спілкуються і працюють в інтернеті. Тому телефонний зв'язок, газети, телебачення відходять на другий план. Величезна кількість людей шукає всю потрібну їм інформацію в мережі.

Тому, створивши свій сайт, будь-яка компанія може приваблювати нових користувачів та підвищувати продажі, або пропонувати інші послуги у вигляді банерної реклами. Вимоги до вигляду та вмісту сайту постійно змінюються, його організація повинна забезпечувати легкість і швидкість модернізації. І в цьому визначальну роль відіграють системи управління контентом [10].

Далеко не завжди є можливість швидко зв'язатися з розробниками проекту, для того, щоб вони забезпечили додавання інформації на ресурс. Крім того, самостійне корпоративне використання контенту сайту та управління ним –

запорука безпеки інформації. З CMS користувач має можливість сконцентруватися саме на утриманні тієї інформації, яку він додає на сайт, а не корегуванні програмного коду.

Актуальність роботи полягає в тому, щоб створити сучасний веб-сайт під назвою «Broker», який орієнтований на розміщення актуальних економічних новин. Справа в тому, що в наш час всі інформаційні видання у вигляді газет та журналів відходять в минуле у зв'язку з розвитком технологій, тому кращим варіантом донести до читачів актуальні новини є веб-сайт.

Метою роботи є розробка елементів веб-ресурсу, які задовольняють технічні вимоги отримані від замовника сайту, наприклад: розміщення котирувань, які будуть оновлюватись в режимі реального часу.

Об'єктом розробки є створення сайту «Broker».

Предмет розробки – методика розробки веб-сайту «Broker», який задовольняє вимоги замовника.

Методами розробки є використання CMS WordPress та декількох плагінів для полегшення та вирішення поставлених задач, використовуються мови програмування PHP, JS та підключення API.

Бакалаврська робота складається з таких розділів:

- Вступ
- Інформаційна частина
- Теоретична частина
- Практична частина
- Висновок

Виконанням бакалаврської роботи є створений та готовий до роботи сайт «Broker».

1. ПОСТАНОВКА ЗАДАЧІ

Перед розробником була поставлена задача замовником проекту створити сайт економічних новин. Перед початком створення веб-сайту розробник отримав технічне завдання та інструменти для його виконання.

Веб-сайт має бути створений на базі CMS WordPress на основі платного шаблону теми представленої замовником. Головними вимогами до сайту має бути чітка структура, гарний дизайн, оптимізація, розміщення блоку новин та відображення котирувань, вартість обміну валют і криптовалют, цін на сировину в режимі онлайн. Для розробника також було поставлене завдання створити та підключити домен сайту та налаштувати веб-хостинг [1].

Додатковим завданням було створити адміністративну частину сайту, яка б давала змогу розділяти рівень прав доступу для редакторів сайту та адміністраторів, також керувати динамічними даними. Сайт має бути відкритим для пошуку, оптимізованим для мобільних пристроїв та сучасних інтернет-браузерів, та використовувати фірмові кольори компанії.

Особливості проектування веб-сайту:

- Під час розробки прототипу сайту важливо дотримуватися єдиної стилістики оформлення елементів.
- Для того щоб зробити прототип сайту необхідно розробити його структуру, де чітко позначити ієрархію блоків, розділів і суброзділів.
- Не рекомендується копіювати подібні елементи, якщо можливо розмістити по рівній інформацію на одній. Наведемо приклад: контакти розміщуються або в розділі про компанію або в нижній частині сторінки, тому виділяти для них сторінку просто не має сенсу.
- Якщо функціонал одного блоку дублює функціонал іншого краще видалити його, оскільки повтори заплутують користувачів ресурсу.
- При створенні прототипу сайту важливо позначити шлях користувача до найважливіших розділах сайту. Створення прототипу сайту передбачає створення

схеми основних сторінок з їх пріоритетністю. Наприклад, якщо фірма не планує часто викладати новини, то немає сенсу верстати для них окрему сторінку [14].

Головною вимогою було, зацентувати увагу не на створенні технічного сайту, а на створенні дизайнерського сайту з гарними візуальними ефектами, тобто задача полягає в роботі з CSS Java Script та плагінами, для кращої візуалізації.

Крім того, сайт має бути створений згідно технічного завдання, та містити наступні сторінки: «Главная», «Про нас», «Новости», «Контакты».

На головній сторінці мають бути блоки новин, плагіни та інформери з курсами валют, динамічні новини та котирування в реальному часі.

На сторінці «Про нас» має міститись інформація про команду редакторів сайту.

Сторінка «Новости» – це окрема сторінка на якій зображено тільки блок новин за сортуванням згідно останнього розміщення новини.

На сторінці «Контакты» – має бути інформація про електронну адресу для зв'язку та форма зворотнього зв'язку.

Сайт має коректно відображатись в сучасних веб-браузерах та використовувати кольори та логотипи компанії.

2. ІНФОРМАЦІЙНА ЧАСТИНА

2.1. Історія розвитку веб-сайтів

Веб-сайт в комп'ютерній мережі – це об'єднана під однією адресою сукупність документів (веб-сторінок) приватної особи або організації. Веб-сайт можна вважати віртуальним офісом, який, як і справжній офіс приймає відвідувачів, користувачів мережі Інтернет. Від того наскільки професійно він буде зроблений, як буде організовано спілкування з користувачем - потенційним клієнтом, залежить успіх маркетингу компанії в Мережі. Історія розвитку сайтів починається з середині 90-х років XX-го століття, коли у Мережі з'явилися перші веб-сайти і складається з декілька етапів [7].

Сайти першого покоління з'явились у Мережі у 1994-1997 роки. Це були текстові, статичні сайти, інформація сайтів була не адаптованою під особливості сприйняття її з екрану, інструментом створення таких сторінок став HTML, без якого в сучасному світі не можливо створити інтернет сторінку власноруч.

Сайти другого покоління з'явились у Мережі у 1998-2002 роки. В даному випадку основою сайту вже була система управління контентом (СУК). Використання СУК означає перехід від статичного подання інформації на сайті до динамічного. Динамічне формування сторінки означає, що до моменту запиту сторінки користувачем Інтернету вона не існує, а формується сервером «на лету» у відповідь на запит. Це дозволяє швидко змінювати зміст і зовнішній вигляд сторінки в залежності від заданих параметрів, одним з популярних на той час систем була Site Core Web CMS, яка на жаль не дійшла до країн СНГ.

На сайті з'явилась спеціальна зона – екстранет, головною метою якої була персоніфікація спілкування з клієнтами: кожний клієнт компанії отримував свій власний «кабінет», в якому йому була доступна вся необхідна ділова інформація, а компанія – власник сайту отримала можливість надавати різним клієнтам різні пропозиції щодо спільного ведення бізнесу. Крім того, у цей час робились перші спроби інтеграції сайту з зовнішніми і внутрішніми програмами. Незважаючи на

суттєвий крок вперед в своєму розвитку, інформація на сайті, як і раніше була не адаптованою під особливості сприйняття її з екрану.

Сайти третього покоління з'явилися у мережі у 2003 – 2005 роках. Тепер основою сайту була промислова система управління контентом, на цьому етапі виникає рішення «інтернет+екстранет+інтранет», тобто на сайті була створена зона, що доступна тільки співробітникам компанії – інтранет. В дизайні сайтів 3-го покоління розпочався пошуку оптимуму, також був продовжений активний розвиток інтеграції з зовнішніми і внутрішніми програмами, прикладом таких програм на той час був 1С створений розробниками із країн СН [9].

Сучасний сайт – це сайт 4-го покоління, яке почало своє існування з 2006 року. Зараз сайт є частиною загальної інфраструктури компанії. Дизайн сайту змінився: ілюстрації є тільки там, де це дійсно необхідно. Сайт повністю інтегрується з внутрішніми корпоративними системами, іншими сайтами і мобільними сервісами, створюючи єдине комунікаційне середовище.

Розвиток сайтів тісно пов'язаний з розвитком самої мережі Інтернет. Особливістю сучасного етапу розвитку Мережі, що має назву Web 2.0, є принцип залучення користувачів до наповнення сайту контентом. Саме за цим принципом створена і постійно розвивається онлайн-енциклопедія Wikipedia. Крім даного принципу, Web 2.0 характеризується появою і розвитком Web-служб, використанням технології AJAX (Asynchronous JavaScript and XML), Web-синдикації, міток, технології mash-up, а також соціалізації Web-сервісів.

Якщо можна було б описати дизайн в стилі Web 2.0 одним словом, то це буде слово «Простота». Сьогодні прості, сміливі та елегантні рішення в дизайні, дозволяють сказати більше, використовуючи менше слів:

- вони дозволяють дизайнерам, переходити відразу до теми сайту, направляти погляд відвідувача на основні елементи, з допомогою декількох добре підібраних візуальних ефектів;
- вони використовують не багато слів, але говорять більше, і ретельно підбирають зображення для досягнення бажаного ефекту;

- вони відкидають ідею про те, що дизайнер не можете передбачити, чого відвідувачі хочуть від вашого сайт [3].

2.2. Основні етапи проектування веб-сайтів

Більша частина населення Землі все ж залишається простими користувачами у всесвітній мережі. Однак, є ті, для кого Internet – це щось більше, ніж засіб пошуку потрібної інформації. Ці люди – веб-розробники.

Веб-розробка для програміста і дизайнера це і хобі, й робота водночас.

Робота, що вимагає постійного розвитку і пошуку нових рішень. Мови програмування постійно вдосконалюються, а програми оновлюються щомісяця (або й щодня). Тож розробникам необхідно постійно тримати руку на пульсі, аби не відстати з огляду на темпи розвитку сучасних технологій, ще пару років тому розробники і не чули про такий інструмент як «Ruby on Rails», а зараз, навіть прості студенти навчальних закладів можуть вивчати його та використовувати в подальшому створенні веб-ресурсів [5].

Розробка для програміста або дизайнера – як творчість для художника. А кожний створений сайт – предмет гордості. Розробник – перш за все, фахівець. Його завдання – не просто писати код. А завдання дизайнера не просто створювати дизайн. Кінцева мета для обох – створення продукту, який повністю задовольняє вимоги клієнта.

Процес розробки чітко розділений на етапи, що повинні бути реалізовані у рамках проекту.

Основні етапи веб-розробки:

- Аналіз брифу і написання ТЗ. Бриф – це такий документ, своєрідна анкета, яку заповнює замовник. У брифі клієнт вказує свої побажання щодо дизайну, функціоналу сайту чи програми та інші деталі проекту;
- Прототипування. Важливий етап, особливо для великих проектів. Прототип – це схема всіх або декількох сторінок сайту у формі ескізу або HTML-

документа, в якому відображені взаємодії та структурні елементи майбутнього сайту: меню, кнопки, форми та інше;

- Наповнення сайту. Цим займається контент-менеджер, власник сайту або ІТ-компанія. Той етап, коли сайт оживає і стає унікальним. Починає “розповідати” заради чого він створений, які товари і послуги можуть бути надані;

- Програмування. Front-end розробники перетворюють макети з дизайном в інтерактивні веб-сторінки за допомогою мови HTML та CSS (по-суті, перетворюють малюнок на код). Back-end розробники поєднують/інтегрують зверстані веб-сторінки з базою даних та системою керування сайтом. Їх задача – зробити сайт не просто закодованою картинкою, а перетворити його на повноцінний функціональний робочий інструмент;

- Розробка дизайну. На даному етапі дизайнер промальовує сторінки ресурсу. Сьогодні важливо, аби сайт мав не лише десктопну версію, а й мобільну. Незмінна вимога до дизайну – “дружній” інтуїтивно зрозумілий інтерфейс «user friendly». Тобто такий, коли користувач легко і швидко знаходить необхідну інформацію на сайті;

- Тестування. Ресурс проходить тестування на відповідність макетів дизайну, швидкість роботи, оптимізацію під мобільні пристрої і відображення в різних браузерах. Фінальна перевірка якості – це оцінка готового продукту з точки зору, як програміста, так і користувача. В великих компаніях в наш час використовують нову ланку фахівців у сфері ІТ, такими фахівцями є «QA тестувальник» головною роботою яких є перевірка працездатності всього функціоналу сайту та правильний і чіткий опис помилок «багів». Тестується працездатність проекту загалом;

- Деплой сайту. На цьому етапі компанія-розробник допомагає клієнту вибрати хостинг (місце в інтернеті) і отримати домен (адреса ресурсу) – два моменти, необхідні для розміщення сайту. Після чого переносять сайт на цей хостинг. Ресурс відкривається для пошукових систем. І проходить етап фінального тестування після переносу;

- Просування ресурсу. SEO-фахівці, SMM-менеджери, таргетологи займаються комплексним просуванням ресурсу: аналізують його роботу, складають стратегію внутрішньої і зовнішньої оптимізації та втілюють її у життя [2].

Отже, в наш час, коли конкуренція на IT ринку та сам ринок постійно є ліквідними та стрімко зростаючими, дуже важливо використовувати «інженерний» підхід в створенні веб-сайтів, лише з огляду на правильно створену концепцію майбутнього сайту вже можна сказати чи буде сайт популярним, сучасним та конкурентоспроможним.

Веб-технології як і все в наш час дуже стрімко розвивається, саме тому постійно треба бути в «тренді» і використовувати більш зручні та сучасні технології розробки, здавалося б, що 10 років назад звичайну інтернет сторінку було створити фактично неможливо для звичайного користувача, потрібно було вивчати HTML, мати вільний доступ до інтернету та бути обізнаним в технологіях розміщення сайту в мережі інтернет, сьогодні навіть учні старших класів в школах можуть буквально в пару кліків миші створити власну веб-сторінку.

2.3. Сучасні веб-технології

Ще кілька років тому створення веб-сайтів було трудомістким процесом, для цього необхідно було володіти відповідною мовою Веб-програмування. На даному етапі є безліч безкоштовних сервісів, які надають уже готові шаблони для створення веб-сторінки, а також безкоштовних веб-хостингів, які дозволяють розмістити сторінку в мережі Інтернет.

Нині спостерігається стійка тенденція застосування таких платформ, що дозволяють створювати й ефективно керувати інформаційним наповненням веб-порталу. Як правило, в основі таких рішень лежать або замовлені веб-розробки, що базуються на серверних технологіях типу ASP, ASP.NET, JSP, PHP, або вже готові потужні засоби для створення веб-сайтів за цими технологіями [6].

В наш час, розвиток веб технологій набрав дуже швидкого розвитку, кожного дня тисячі розробників по всьому світу використовують свої знання та час на розробку різного виду інструментів для інших розробників, аби встигати за сучасним трендом ринку "швидко, бюджетно, якісно".

З кожним роком кількість додатків та інструментів для веб розробників зростає на сотні та навіть тисячі одиниць, сьогодні хочу поговорити про останні розробки в web-технологіях, такі як технологія API, та самі популярні бібліотеки для web-розробників: JQuery, Bootstrap.

Для початку поговоримо про те, що собою являють ці інструменти, та які переваги вони надають в порівнянні з стандартними "застарілими" інструментами розробки .

API (Application Programming Interface) — це набір готових класів, процедур, функцій, структур і констант, що надаються додатком (бібліотекою, сервісом) для використання в зовнішніх програмних продуктах. Своїми словами, API надає нам можливість використовувати чужі напрацювання в своїх цілях. Отже, застосування API програмістам спрощує створення коду.

Для інтерактивної та активної взаємодії користувача з web-додатком використовується мова програмування java script. Java script — це динамічна, об'єктно-орієнтована мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується як частина браузера, що надає можливість коду виконуватися на стороні клієнта, таким чином знімаючи навантаження з основного сервера на якому і виконується основний функціонал системи.

Ця мова також може використовуватися для програмування на стороні сервера, розробки ігор, стаціонарних та мобільних додатків, сценаріїв в прикладному, всередині PDF-документів тощо. JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією [16].

Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування і деякі відповідні архітектурні властивості, зокрема: динамічна та

слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Jquery – це популярна JavaScript-бібліотека з відкритим кодом. Синтаксис jQuery розроблений, щоб зробити орієнтування у навігації зручнішим завдяки вибору елементів DOM, створенню анімації, обробки подій, і розробки AJAX-застосунків. jQuery також надає можливості для розробників, для створення плагінів у верхній частині бібліотеки JavaScript. Використовуючи ці об'єкти, розробники можуть створювати абстракції для низькорівневої взаємодії та створювати анімацію для ефектів високого рівня.

Це сприяє створенню потужних і динамічних веб-сторінок. Bootstrap – це CSS/HTML-фреймворк для створення сайтів. Іншими словами, це набір інструментів для верстки. У ньому є ряд переваг, завдяки яким BS вважається найпопулярнішим серед собі подібних.

Переваги Bootstrap:

- Швидкість роботи – завдяки безлічі готових елементів верстка з бутстрапа займає значно менше часу;
- Масштабованість – додавання нових елементів не порушує загальну структуру документів;
- Легко налаштовувати – редагування стилів проводиться шляхом створення нових css-правил, які виконуються замість стандартних. При цьому не потрібно використовувати атрибути типу “! important”;
- Велика кількість шаблонів;
- Величезне співтовариство розробників;
- Широка сфера застосування – Bootstrap використовується в створенні тем для практично будь-яких CMS (OpenCart, Prestashop, Magento, Joomla, Bitrix, WordPress і будь-які інші), в тому числі для односторінкових додатків.
- Особливою популярністю користується Bootstrap для створення односторінкових або лендингів (landing page) [13].

Перерахувавши всі переваги цих інструментів, можна зрозуміти, що в наш час достатньо оволодіти сучасними інструментами які дають змогу спрощувати створення веб-ресурсів.

Якщо більш просто розглянути інструменти які ми вже перерахували, то простими словами можна пояснити роботу кожного з них на прикладі популярних ресурсів.

Наприклад зараз дуже важко знайти хоча б один сайт, який би в своїй роботі не використовував бібліотеку JQuery яка пропонує готові рішення взаємодії візуальних ефектів, Bootstrap свою чергу дає нам змогу використовувати універсальні стилі, без потреби писати стилі самому, достатньо їх просто редагувати та "підігнати" за власним бажанням та потребою.

Стосовно API технологій все набагато цікавіше, якщо уявити себе викладачем, який пояснює що таке API можна пояснити як уявний міст, через який можна передавати, або отримувати необхідну інформацію в режимі онлайн, тобто міст зв'язує між собою ваш ресурс та сервіс або ресурс який вам надав доступ до API, наприклад котирування валют, або індекси бірж працюють на технології API.

Спочатку котирування розраховуються на стороні серверу біржі, передаються регулятору, після чого за допомогою API – котирування з усіма правками потрапляють до користувача на будь який ресурс, де було встановлено даний API, це свого роду оптимізація процесу, тому що розрахунки на себе беруть сервери бірж під наглядом професіоналів та гідним захистом, а ми лише можемо вивести цю інформацію на власних ресурсах [8].

Отже, слідкувати за сучасними технологіями вкрай важливо, це спрощує розробку кінцевого продукту, та дозволяє заощадити кошти на розробці.

Використовуючи сучасні інструменти розробки ви, як фахівець одразу можете себе показати в будь якій IT компанії і стати провідним фахівцем, шкода що в сучасній системі освіти дуже мало часу витрачають на розгляд таких інструментів розробки які дуже актуальні в наш час. Для більшого розуміння проблеми, сайти створені за допомогою лише стандартних HTML, PHP, CSS та JS, без додаткових фреймворків та без сучасних бібліотек займають дуже багато часу, та економічно не вигідні.

2.4. Етапи розробки веб-сайта

2.4.1. Вибір програмних засобів для розробки веб-сайта

В період з 2015 по 2020 рр. напрямок веб-розробки, як окрема ланка в it-ринку розділився на декілька чітких груп. Кожна група потребує кваліфікованих фахівців, які здатні працювати з сучасними програмними засобами розробки. Проаналізуємо та визначимо кожен з цих груп в логічній послідовності:

- Веб-дизайнери. З їх роботи розпочинається створення професійних сайтів. Творчий підхід до дизайну сайту – це 50% успіху майбутнього проекту. У своїй роботі дизайнери використовують безліч необхідних їм інструментів, як провірених часом, так і сучасних розробок. Наприклад, вічна класика – це “Photoshop”, сучасний Adobe Illustrator.
- Front-end – продовжують роботу веб-дизайнера, роблячи з шаблону фізичний сайт. Використовують Js, JQuery, Css, та редактор коду Visual Code, або Sublime Text.
- Back-end – користуючись мовою програмування PHP, C#, Java – наповнюють сайт функціоналом та працюють над його захистом
- SEO – фахівці працюють з наповненням сайту, html кодом та оптимізують його роботу для пошукових систем.

Мова HTML є базовою в області технологій створення сайтів, оскільки відносно легка в освоєнні. Але надмірна простота є і її недоліком. HTML (від англійського Hyper Text Markup Language – мова розмітки гіпертексту) чудово відповідає вимогам раннього періоду розвитку технологій створення сайтів, але з подальшим його розвитком виникли істотні проблеми. HTML надає такі можливості:

- Видавати мережеві документи із заголовками, текстом, таблицями, списками, фотографіями і тому подібне.

- Отримувати інформацію з мережі через посилання гіпертексту при натисканні кнопки.
- Створювати форми для посилки запитів на видалені комп'ютери, щоб здійснювати пошук інформації, здійснювати бронювання, замовляти товари і тому подібне.
- Включати електронні таблиці, відео кліпи, аудіо кліпи, і інші програмні застосування безпосередньо в їх документи [11].

PHP – це скриптова мова програмування, створена для генерації HTML-сторінок на Webсервері і роботи з базами даних. На даний момент він підтримується практично усіма представниками хостингу, входить в "стандартний" набір для створення сайтів.

Головним чином, PHP застосовується при написанні скриптів, що працюють на стороні сервера; таким чином, PHP здатний виконувати все те, що виконує будь-яка інша програма CGI (наприклад, обробка цих форм, генерувати динамічні сторінки, посилати і приймати cookies).

Переваги PHP:

- доступний для більшості операційних систем, включаючи Linux, багато модифікацій Unix (такі, як HP - UX, Solaris і OpenBSD), Microsoft Windows, Mac OS X, RISC OS, і багатьох інших;
- включає підтримку більшої частини веб-серверів (для більшості серверів PHP постачається як модуль, для інших, що підтримують стандарт CGI, PHP може функціонувати як процесор CGI);
- підтримує великий круг баз даних;
- підтримує DBX для роботи на абстрактному рівні (таким чином можна працювати з будь-якою базою даних, використовуючих DBX); ODBC (тобто можна працювати з будь-якою базою даних, що підтримує цей стандарт);
- підтримує "спілкування" з іншими сервісами з використанням різних протоколів: LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (на платформах Windows), а також багатьох інших;

- підтримує стандарт обміну складними структурами даних WDDX;
- підтримує об'єкти Java, дає можливість використовувати їх як об'єкти PHP .

В Україні маленькі IT-компанії потребують від кандидатів на посаду Junior знання всіх цих інструментів хоча б на базовому рівні, тим самим заощаджують на їх утриманні, це не вирішує багато проблем, адже краще бути фахівцем в чомусь одному, чим аматором в усьому. Фахівця який досконально вміє користуватись всіма інструментами розробки називають FullStack розробник [12].

Отже, неможливо зробити хороший сайт, не знаючи декількох мов програмування, на яких базуються сучасні веб-технології. І використання тієї, або іншої технології, а також навіть декількох залежить від призначення веб-ресурсу, який розробляється.

2.4.2. Обґрунтування вибору системи управління вмістом

CMS (“Content Management System”, що означає “Система керування вмістом”). Такі системи також базуються на HTML, CSS та різноманітних мовах програмування, що згадані раніше, але користувачеві пропонуються вже готові рішення, які достатньо встановити та провести налаштування і наповнити контентом відповідно до своїх потреб.

Це досить зручно, адже не потрібно займатись самотійною розробкою всього необхідного функціоналу чи замовляти сайт в компаній, що займаються веб-розробкою. До найпопулярніших CMS відносять: WordPress, Drupal, Magento, Joomla, OpenCms, phpBB та інші.

CMS можуть бути цікавий різним структурам: від компаній, яким потрібно підтримувати свій сайт, до студій, що займаються розробкою сайтів:

- CMS не вимагають професійної технічної підготовки при використуванні. Практично будь-який користувач може працювати з CMS і управляти вмістом сайту: додавати і видаляти статті, різні модулі (стрічка новин, форуми, гостьові книги, статистика) – тобто все те, що робить сайт сайтом;

- CMS можуть бути успішно інтегрований у внутрішньокорпоративну інформаційну систему і служити для організації документообігу;

- CMS може працювати в зв'язці з системою документообігу, що особливо зручно, коли сайт розглядається не просто як форма реклами і присутності в інформаційному просторі, а як свого роду електронного офісу: певна кількість інформації, при її оновленні "усередині" компанії, може тут же відображатися на сайті;

- CMS може бути використаний як інструмент при наладці сайту як каналу взаємостосунків з клієнтами компанії – в цьому випадку доцільно зв'язати CMS і CRM-систему компанії. Ця зв'язка відкриває нові можливості в області підтримки клієнтської "відданості": інформація на сайті мобільна, можливий доступ до окремих, призначених для певних клієнтів розділом по пароллю, що надзвичайно зручно у разі чіткої сегментації клієнтів;

- CMS дозволяє здійснювати доступ до редагування інформацією різними адміністраторами, що особливо зручно у разі, коли компанія володіє цілою сіттю дилерів, але одним корпоративним сайтом. Можливо чітке розмежування прав на редагування окремих блоків різними людьми;

- При цьому адміністратори можуть фізично розташовуватися в різних точках земної кулі. Таким чином, ви можете налагодити збір інформації на сайті з багатьох джерел (дилери можуть, наприклад, вносити результати своїх продажів; видалені склади розміщувати інформацію про наявність товарів на складі). Ви одержуєте в свої руки автоматизований інструмент підтримки актуальної інформації на сайті, розподіляючи роботу по наповненню інформацією між своїми філіалами, відділами;

- CMS економічно доцільні при організації внутрішньокорпоративних систем і інформаційних порталів.

CMS рішень багато, але при виборі варто спиратися на наступні можливості:

- призначений для користувача сервіс – наявність тих або інших функцій і модулів, зрозумілість і доступність користувачу;

- технологічність – використання технологій, що дозволяють підвищити пропускну спроможність і надійність системи;
- сумісність (апаратна і програмна) – можливість функціонування системи на різних платформах, сумісність з СУБД, можливість підключення додаткових модулів;
- масштабованість – можливість розвитку і нарощування системи.

Насправді, далеко не будь-яка контент-система виявляє собою готовий програмний продукт. Це може бути всього лише набір різнорідних модулів або ж варіант, створений по індивідуальному замовленню [15].

Таким чином, по ступеню готовності контент-системи діляться на наступні різновиди:

- розробник сам встановлює і настраює контент-систему на сервері замовника;
- контент-система проектується і розробляється під кожний окремий проект і встановлюється розробником.

Використання CMS спрощує та прискорює розробку будь-якого веб-сайту, незалежно від його напрямку та потреби в цілому. Користуючись CMS, замовник економить власні кошти та час, бо є дуже багато готових рішень, для будь-яких потреб, як безкоштовних, так і платних з постійною підтримкою від розробників.

Користуючись платними плагінами, або шаблонами, ми можемо розраховувати на безпеку нашого сайту, та оновлення модулів і отримувати інструкції з правильного використання та налаштування від розробників цих плагінів, або шаблонів.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Розробка веб-дизайну сайту

Веб-дизайн (англ. web design) – галузь веб-розробки, що охоплює цілий ряд напрямів і дисциплін із створення та супроводу сайтів або веб-застосунків, таких як графічний веб-дизайн, проектування інтерфейсів, авторинг (у тому числі стандартизований код і власницьке програмне забезпечення), використання та оптимізація для пошукових систем.

Сучасний веб-дизайн починається з поділу оформлення сайту і змісту. Завдяки такому підходу вносити зміни в зміст сайту, не зачіпаючи його дизайн, або ж змінювати дизайн, не змінюючи зміст, стало набагато легше. Багато в чому цьому сприяють сучасні системи управління контентом – CMS.

Основи веб-дизайну сайту закладаються на етапі розробки технічного завдання – документа, що описує вимоги щодо візуального представлення та структури сайту. У технічному завданні також найчастіше описуються побажання щодо програмної реалізації сайту.

Після розробки та затвердження техзавдання починається етап розробки дизайну сайту. І перше, що необхідно зробити – це створити модульну сітку сайту. Модульна сітка описує розташування елементів на сторінках сайту.

Модульна сітка будь-якої web-сторінки містить, як мінімум, два блоки: для основного контенту сайту і для меню. Якщо на сторінці потрібні додаткові елементи, наприклад, ще одне меню, «підвал» («footer») або «хедер» («header»), то модульна сітка розбивається ще на кілька блоків.

Розміри блоків модульної сітки підбираються індивідуально і можуть бути якими завгодно. Проте слід враховувати, що відповідно до вимог «usability», поява горизонтальної смуги прокручування при перегляді сторінки неприпустимо – це негативно позначається на сприйнятті сайту відвідувачами. Тому в сукупності ширина блоків не повинна перевищувати максимальної ширини екранів представників цільової аудиторії сайту.

У блоці «хедер» прийнято вказувати назву сайту, яке може бути вибрано відповідно до назви компанії або ім'ям автора сайту, а також при необхідності логотип сайту. Основний блок меню в європейських сайтах традиційно розташований зліва, відповідно до напрямку писемності (зліва-направо). Також основне меню може бути розташоване зверху, під «хедером» [17].

Якщо сайт розрахований для перегляду на екранах з різним дозволом, то веб-дизайн, як правило, виконується «гумовим», або ж центрується по горизонталі.

Коли інформаційна модель майбутнього сайту готова, можна братися за художню частину веб-дизайну. Зазвичай дизайн сайту виконується у вигляді шаблонів – певних наборів елементів і їх взаємозв'язків. Саме шаблони дозволяють відокремити розробку візуального дизайну сайту від змісту.

Розрізняють шаблон головної і шаблони типових сторінок сайту. Шаблон головної сторінки оформляється, як правило, більш «витіювато», а шаблони решти сторінок сайту лише підтримують основну ідею.

Розроблений в графічному редакторі шаблон верстається у html-файл. За допомогою тегів мови html і таблиць стилів css задаються всі необхідні параметри сторінки: розміщення елементів, колір і розміри шрифтів, колір фону і т.д.

Робота над шаблоном сайту вимагає обов'язкового тесту на крос-браузерності, тобто на сумісність з різними браузерами і їх версіями. Різні браузери можуть по-різному інтерпретувати код html, тому таке тестування необхідне, щоб уникнути можливого «розвалу» дизайну.

У разі використання CMS, розроблені шаблони «закачуються» у відповідну папку на сервері і налаштовуються розробником для коректного відображення сайту в мережі інтернет, для більшості шаблонів потрібно встановлювати плагіни «пейдж-білдери» або інші плагіни-конструктори сторінок, це дасть змогу в майбутньому людям які не розуміються в програмуванні власноруч створювати елементи сторінок і повноцінні сторінки, та наповнювати їх контентом. Складною частиною налаштування «чужих» шаблонів є те, що під технічне завдання потрібно залишати лише необхідні сторінки, мати певний функціонал якого може бути за замовчуванням, або дуже мало, або занадто багато, тому розробник з готового

шаблону має «вирізати» все, що не підпадає під технічне завдання, а все інше налаштувати. В сучасному ІТ – є ціла ланка таких розробників, їх називають згідно назви CMS яку вони використовують, наприклад: WordPress – developer [18].

3.2. Структура сайту

Логічна структура сайту – подібний набір тематичних рубрик з розподіленими по відповідних розділах документами і наперед спроектованими гіперзв'язками між всіма сторінками ресурсу. Тому, розглянемо логічну структуру на прикладі сайту «Financebroker» (рис. 3.1).

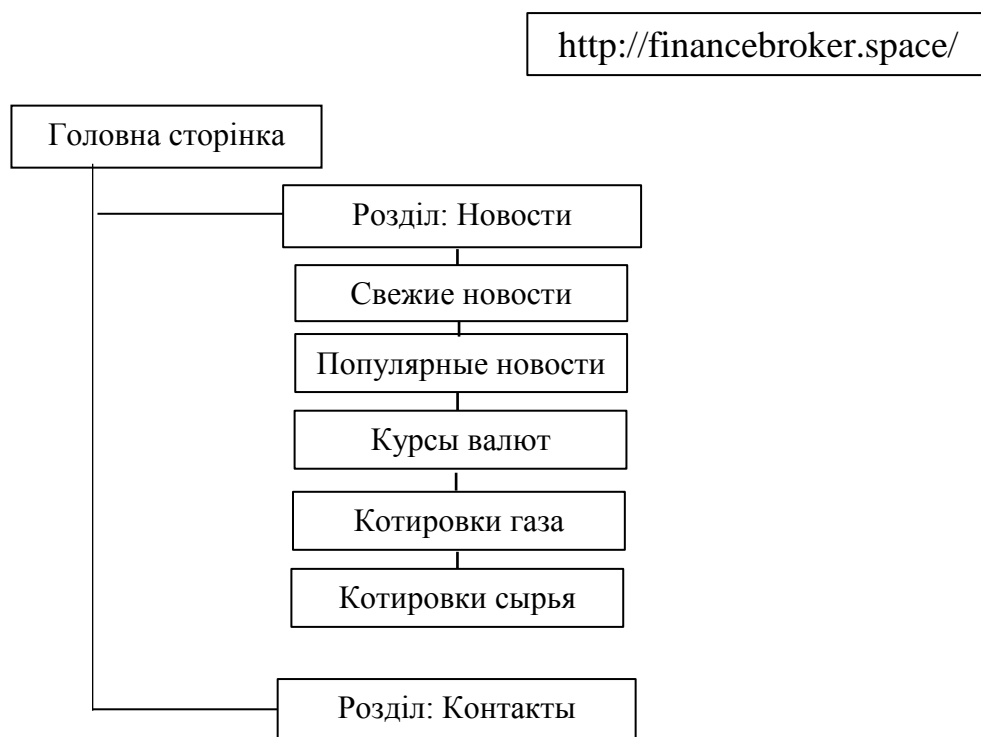


Рис. 3.1 – Логічна структура «Financebroker»

Фізична структура, навпаки, має на увазі алгоритм розміщення фізичних файлів по підкаталогам каталогу, в якому опублікований ваш сайт. Приклад спрощеної фізичної структури одного і того ж ресурсу Інтернету подано на рис 3.2.

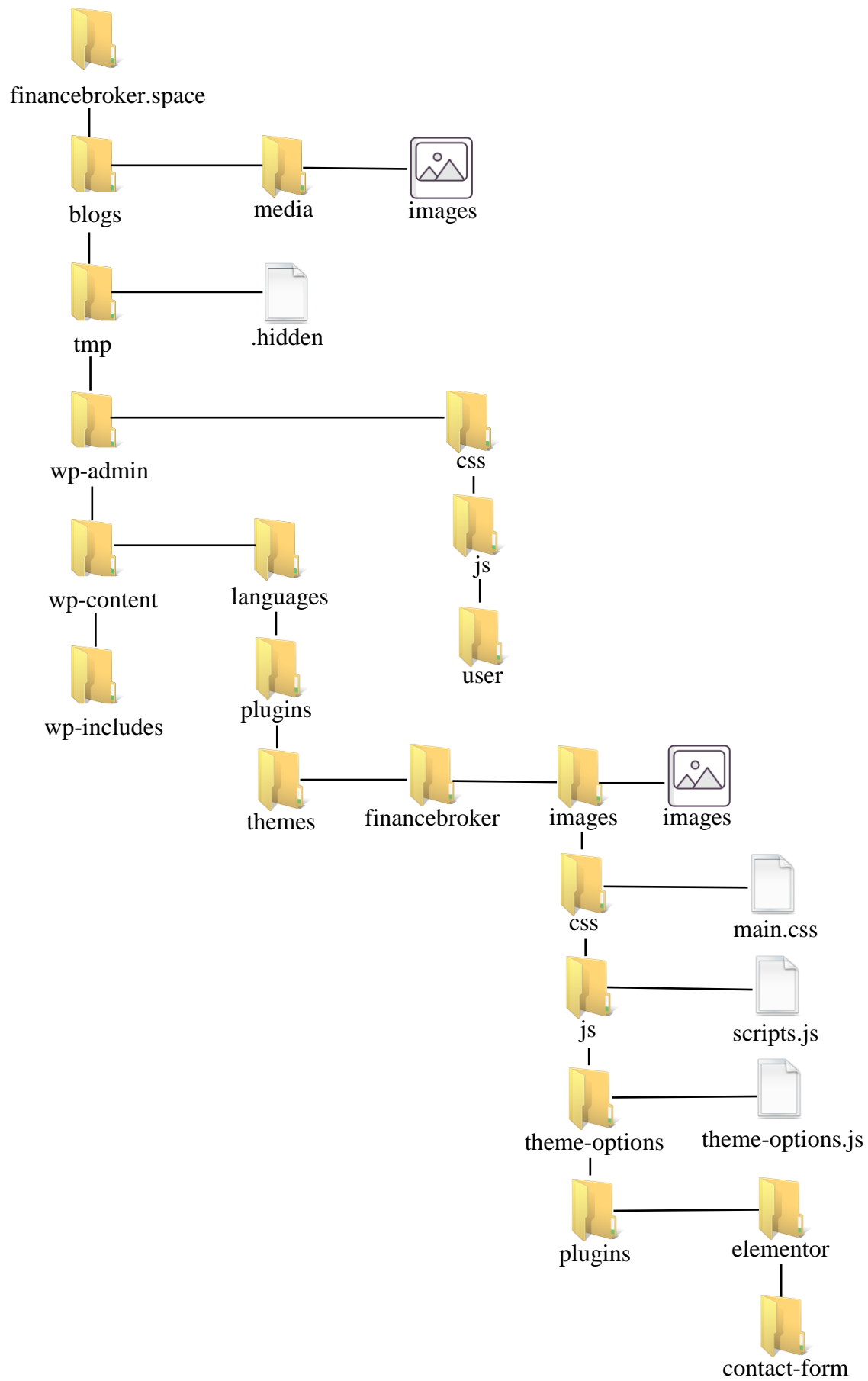


Рис. 3.2 – Фізична структура «Financebroker»

3.3. Контент та його наповнення

Наповнення сайту контентом – фінальний етап у створенні сайту перед тим, як його оцінять реальні відвідувачі онлайн. Саме на цьому етапі важливо сконцентрувати всі зусилля на створенні якісного продукту, на підставі якого будуть судити про ресурс в цілому.

Принципово важлива при контентному наповненні сайту унікальність і корисність матеріалів, які ви розміщуєте на сайті. Тільки ресурс, який пропонує родзинку корисності на своїх сторінках буде дійсно успішним. Не можна забувати, що сайт наповнюється інформацією не для того, щоб було легше привести на нього відвідувачів з різних джерел, але для того, щоб ті, хто прийшов на сайт, вирішили на ньому залишитися.

Заповнення сайту має носити постійний характер, так як повинен стабільно вестися блог, наповнюватися новинами, та оновлювати застарілий контент на основних посадочних сторінках.

Етапи наповнення сайту контентом:

- Складається ТЗ по створенню текстових, фото-, відеоматеріалів;
- Матеріали готуються;
- Готові матеріали проходять первинну вичитку / перевірку на якість, унікальність;
- Унікальні матеріали оптимізуються технічно і програмно під поточні вимоги пошукових систем;
- Матеріали затверджуються з замовником;
- Затверджені матеріали розміщуються на сайті.

Отже, наповнення сайту контентом, самий трудомісткий і дуже відповідальний процес. Будь-яку інформацію у вигляді текстів, зображень, відео, аудіо та інше, називають одним словом – контент. Відвідувачі на сайт заходять саме заради контенту, який там знаходиться. Пошукові системи в своєму топі показують посилання на сайти аналізуючи саме якість контенту сторінок цих сайтів.

Розглянемо наповнення сайту новинами на прикладі сайту «Financebroker» (рис. 3.3).

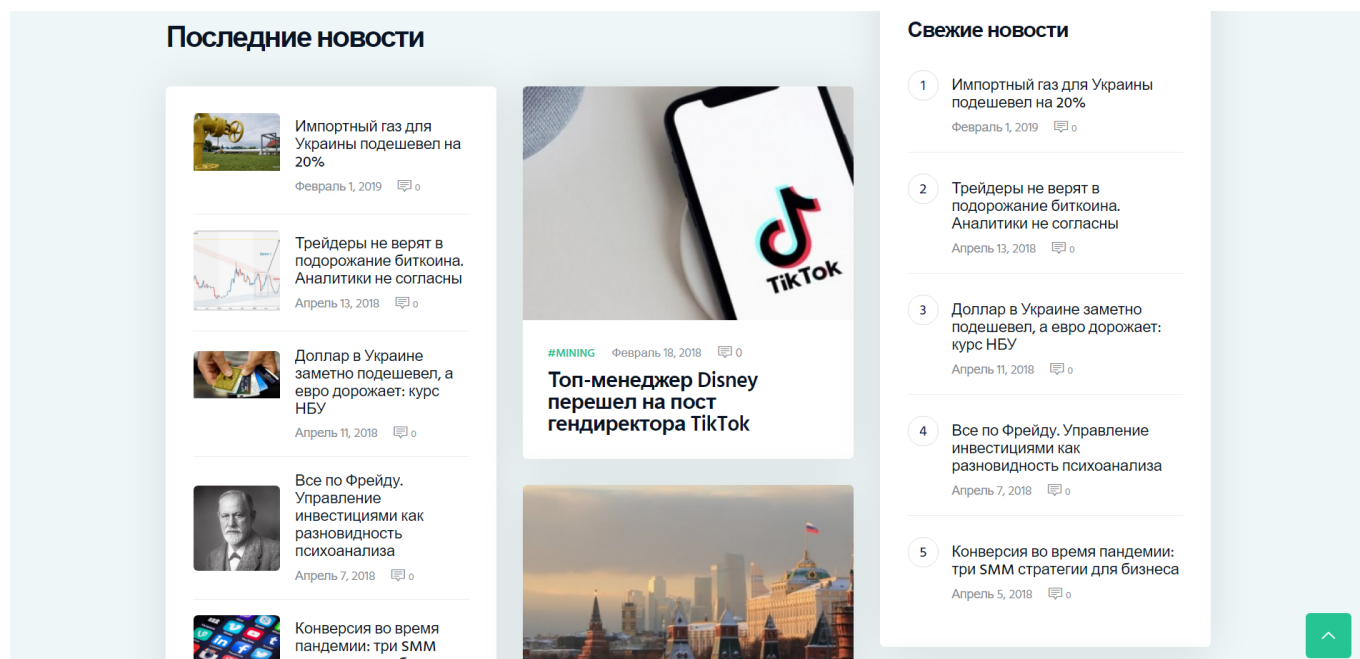


Рис 3.3 – Наповнення контентом

В нашому випадку, наповнення контентом є розміщення актуальних новин у фінансовій сфері, які поділяються на декілька підрозділів, такі як: «Актуальні новини», «Останні новини», зі змогою відкрити кожен з новин та ознайомитися з ними, як зображена на рис. 4.4.

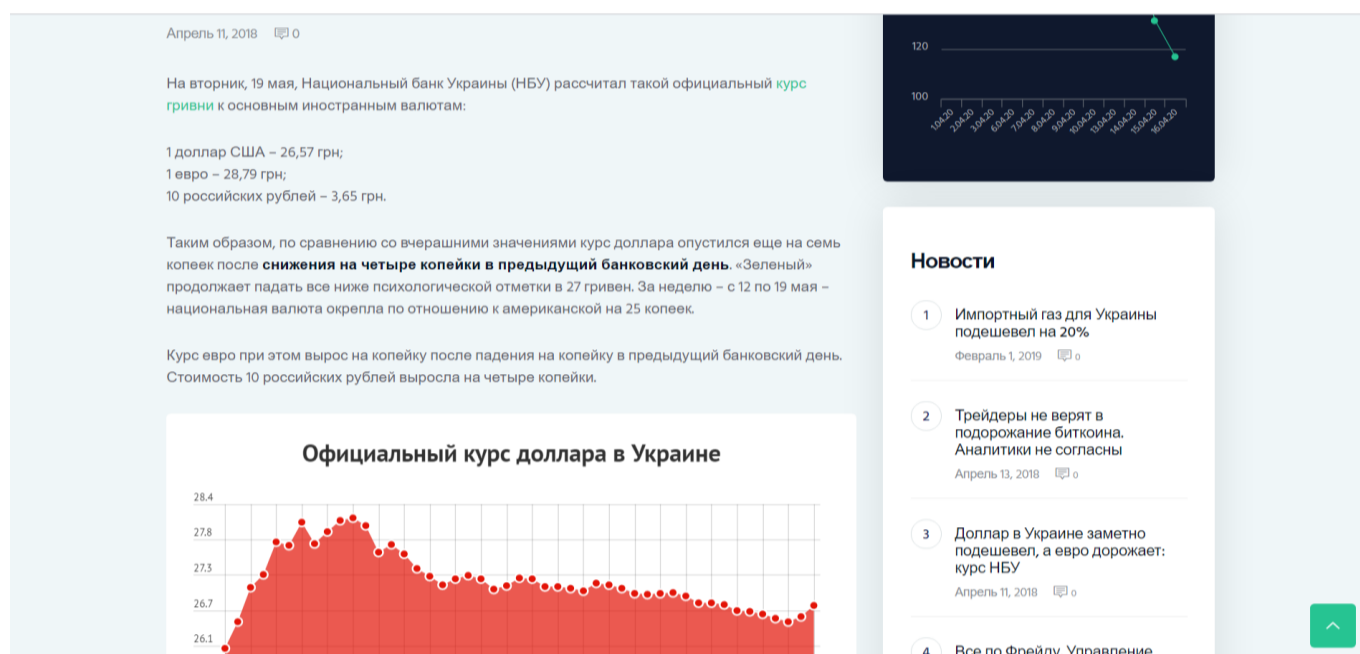


Рис. 3.4 – Сторінка з новинами

Також, приємним та зручним інструментом перегляду в якості додаткового контенту є повністю автоматизовані котирування у вигляді таблиці, та у вигляді графіку, як зображено на рис. 3.5.

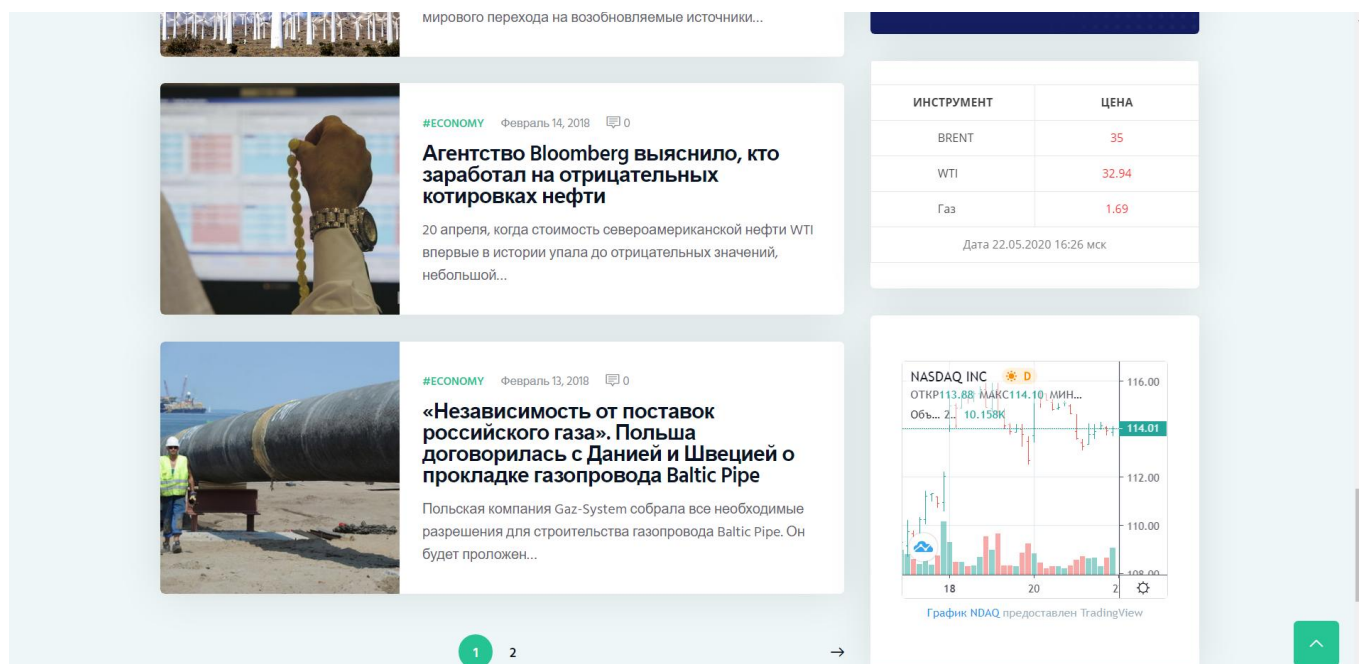


Рис. 3.5 – Додатковий контент

Користувачам може стати цікавою інформація про редакцію блогу з новинами, яка представлена на окремій сторінці «Про нас», як зображено на рис. 3.6.

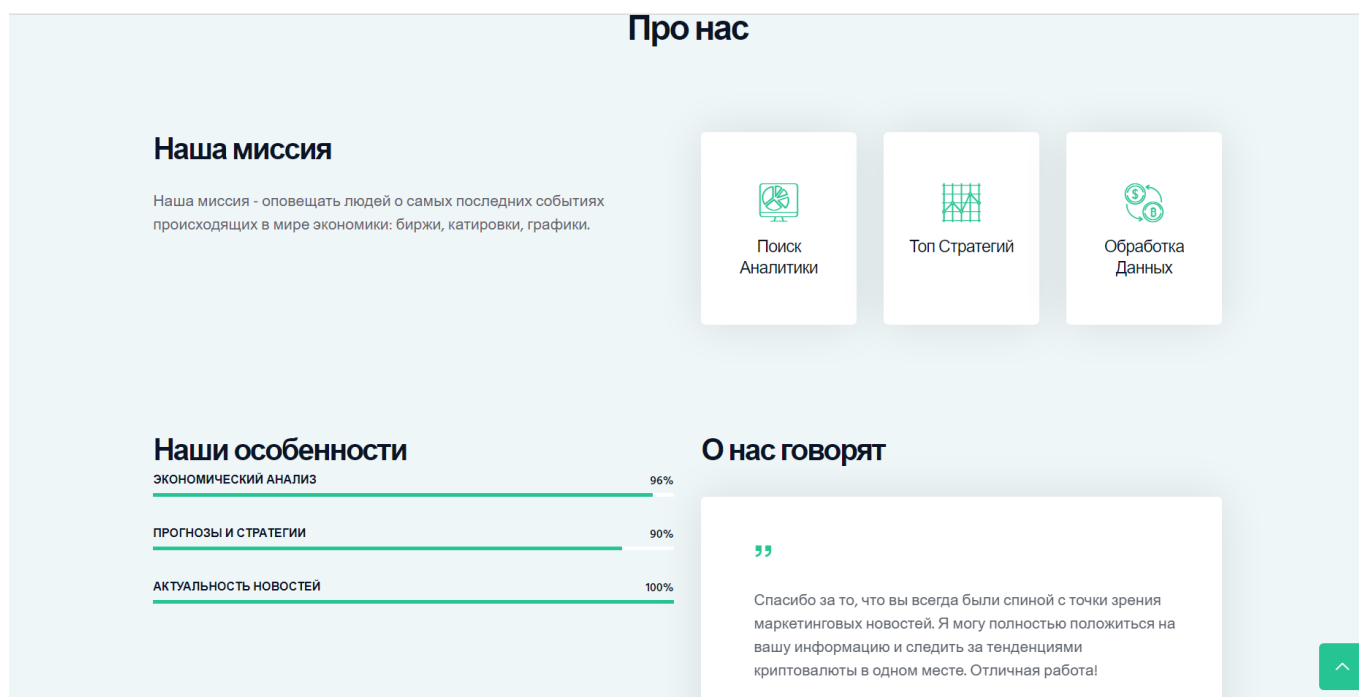


Рис. 3.6 – Сторінка інформації про блог

4. ПРАКТИЧНА ЧАСТИНА

4.1 Встановлення на сервер CMS WordPress

Встановити WordPress, як і будь-який інший веб-скрипт з великої колекції скриптів можна прямо з панелі ISPmanager.

Необхідно перейти в пункт меню "Каталог Web скриптів" панелі ISPmanager та натиснути кнопку «Встановити», як зображено на рис. 4.1.

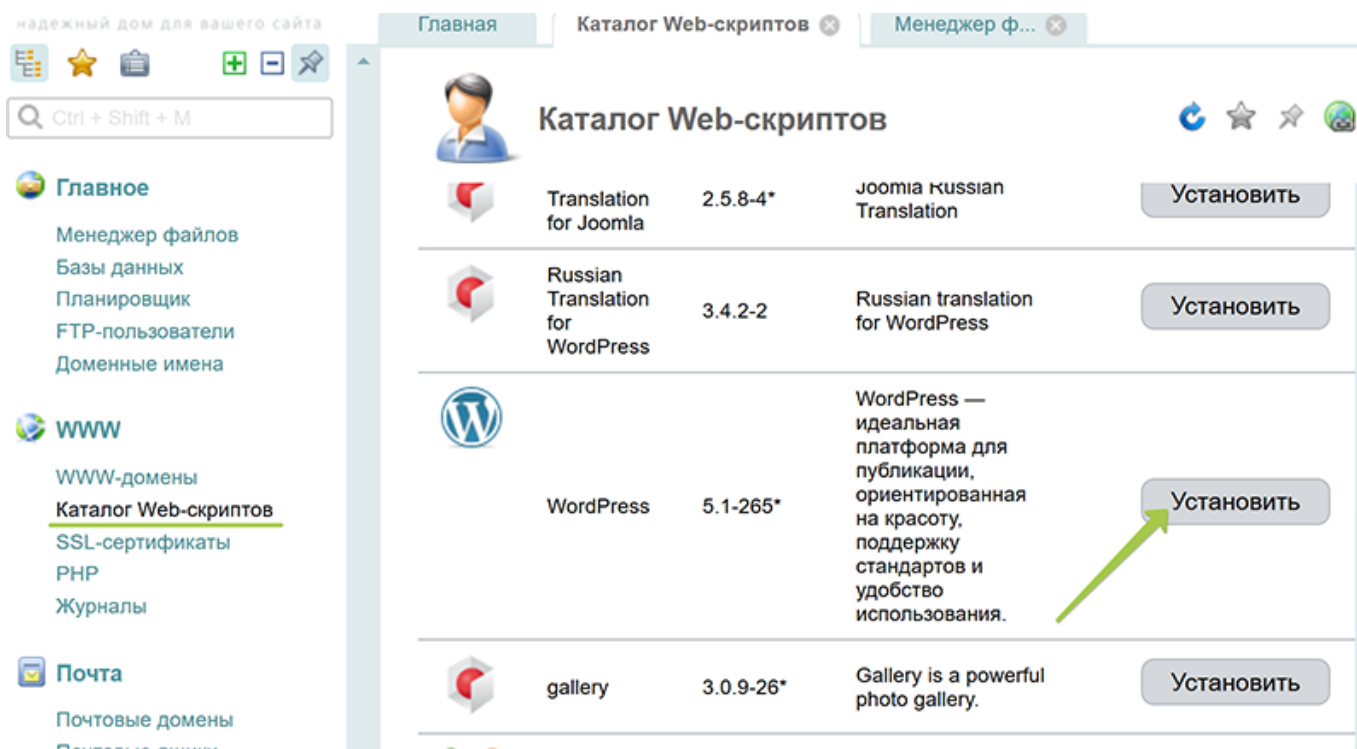


Рис. 4.1 – Початок встановлення WP

Другим етапом необхідно вибрати найновішу версію WordPress із запропонованих в випадаючому списку і натисніть кнопку "Далі".

Потім обрати зі списку попередньо створений WWW-домен. Директорію залишити за умовчанням кореневу – /, якщо потрібно встановити WordPress в піддиректорію сайту наприклад – blog, то вибираємо попередньо для цього створену директорію. На цьому ж кроці вибираємо сервер MySQL, можна залишити за замовчуванням. В кінці натиснути кнопку "Далі" (рис. 4.2).

надежный дом для вашего сайта

Главная Web-скрипт - WordPress

Web-скрипт - WordPress

Шаг 1 Web-скрипт → Шаг 2 Лицензионное соглашение → Парам

WWW-домен ?

Директория *

Сервер ?

Далее >> Отмена

Рис. 4.2 – Другой этап встановлення WP

Третій етап:

- Заповнити всі необхідні поля.
- Придумати унікальні ім'я і користувача бази даних і згенерувати для них пароль.
- Натиснути кнопку "Далі" (рис. 4.3).

WordPress → Шаг 3 Параметры установки → Шаг 4 Подтверж

База данных ?

Имя базы данных *

Пользователь ?

Имя пользователя *

Пароль *

Подтверждение *

Сложный

Рис. 4.3 – Третій етап встановлення WP

Далі необхідно:

- Придумати ім'я і користувача адміністратора майбутнього сайту на WordPress і згенерувати для нього пароль.
- Придумати та встановити ім'я сайту .
- Обрати мову інтерфейсу .
- Встановити електронну адресу адміністратора для відновлення паролю, або логіну (рис. 4.4).

Рис. 4.4 – Четвертый этап встановлення WP

Після всіх налаштувань система повідомить нас про наявність помилок, або про успішне встановлення CMS, як зображено на рис. 4.5.

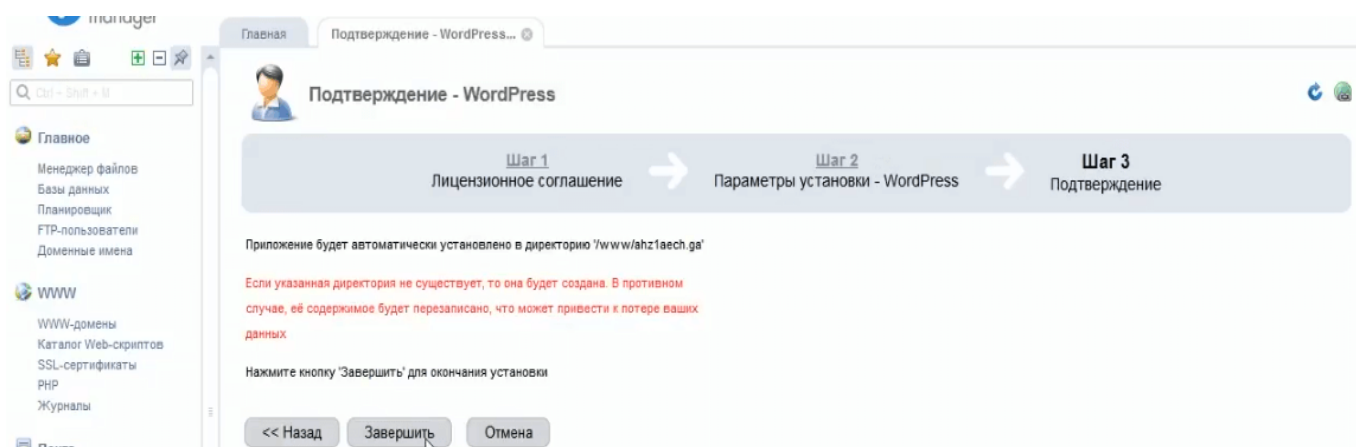


Рис. 4.5 – Завершения установки

4.2. Встановлення платного шаблону сайту

Перед нами була поставлена задача, створити сторінку – інформаційного ресурсу, на основі отриманого від замовника придбаного шаблону сайту формату «Блог». Спочатку може здаватися, що налаштування навіть платних шаблонів – це досить проста річ, але це не зовсім так. Справа в тому, що для налаштування шаблону під технічне завдання замовника є не просто встановити тему, та відкрити сайт для пошукових ботів, а видалити з шаблону весь непотрібний функціонал, встановити необхідні плагіни та бібліотеки.

Складність полягає в тому, що сама структура, як теми, так і WP дуже велика та заплутана, і більшу частину часу розробника витрачається саме на пошук необхідного коду, який необхідно змінити, або видалити, також налаштування необхідних плагінів, хостингу та робота з ними, тому розглянемо спрощену модель встановлення шаблону, за допомогою FTP, як зображено на рис. 4.6.

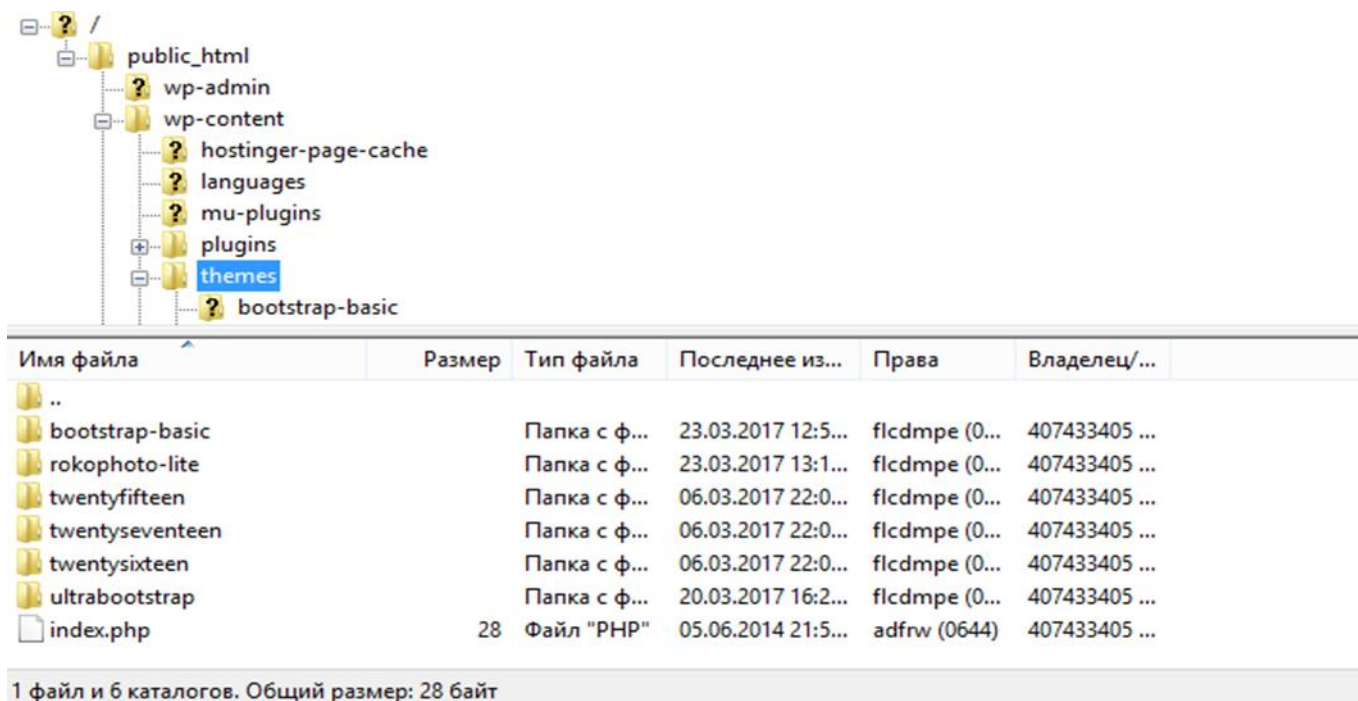


Рис. 4.6 – Встановлення теми за допомогою FTP

Для початку необхідно отримати логін і пароль з хостингу, та вказати його при вході FTP Manager, після чого ми отримаємо список усіх файлів, розміщених на

хостингу, тобто встановлений WP, з його каталогами. Далі нас цікавить каталог «themes», в який і необхідно завантажити архів з темою, як зображено на рис. 4.7.

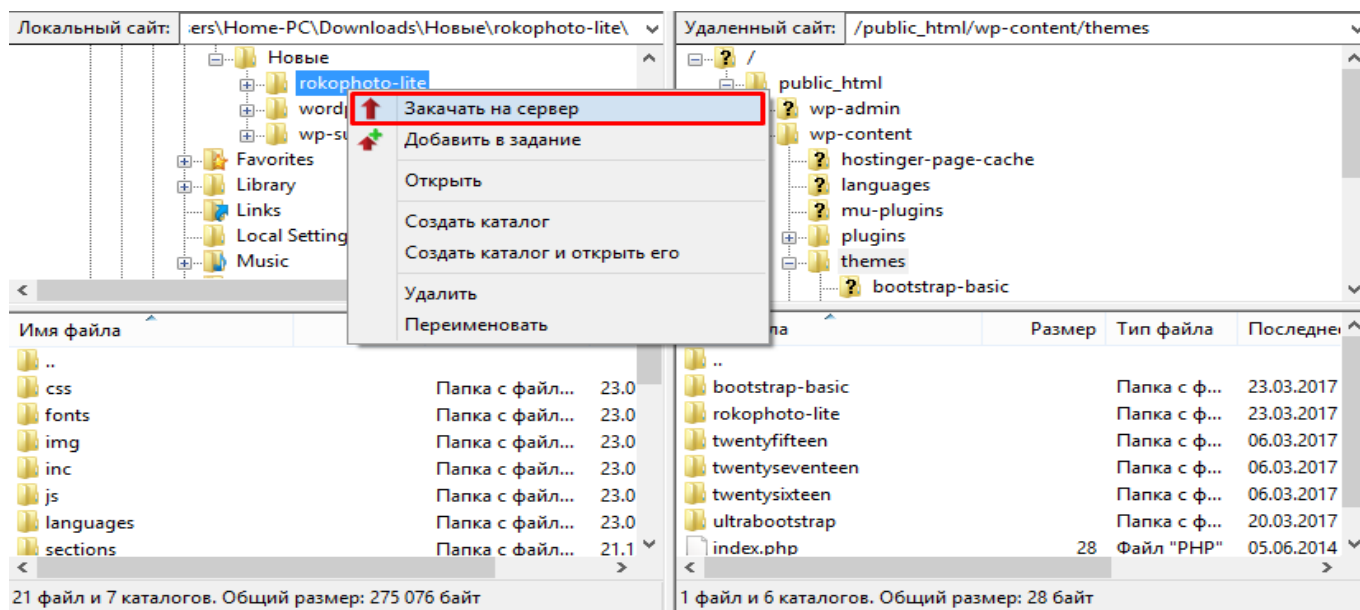


Рис. 4.7 – Завантаження архіву з шаблоном

Після вдалого завантаження нам необхідно перейти в адміністративну частину сайту. Для цього потрібно перейти в «Зовнішній вигляд», підрозділ «Теми», та обрати завантажену тему, натиснути кнопку «Активувати», як зображено на рис. 4.8.

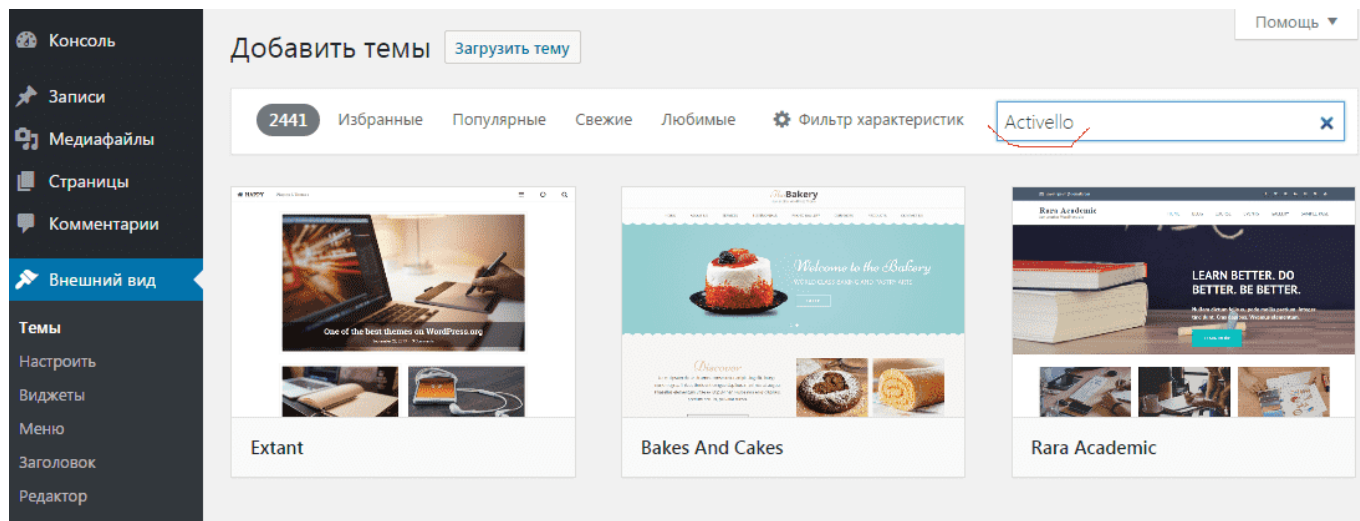


Рис. 4.8 – Останній етап завантаження теми

Якщо встановлення завершилось правильно, необхідно провести встановлення плагінів та перейти на головну сторінку, щоб побачити результати, як зображено на рис. 4.9.

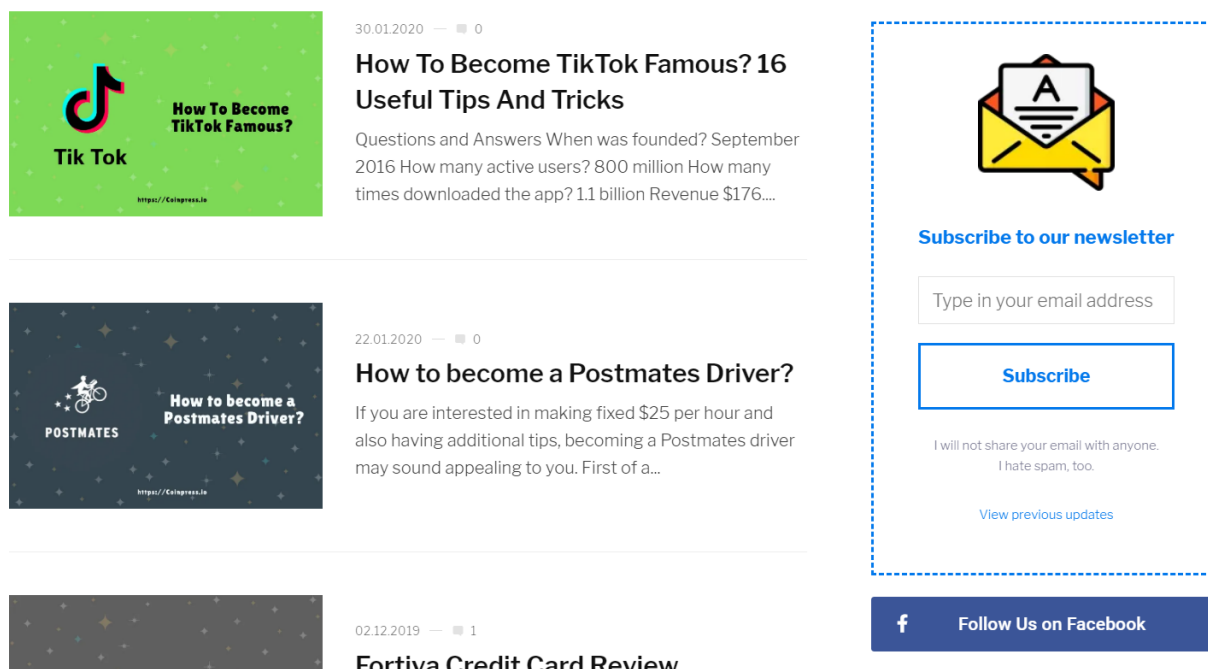


Рис 4.9 – Вигляд неналаштованої теми

4.3. Встановлення необхідних плагінів на сайт

Налаштування будь-якої користувацької теми, розробленої сторонньою компанією, або розробником включає в себе використання деяких додаткових інструментів. Такими інструментами розробки є плагіни, які створюють лише для однієї цілі – збільшити технічні можливості сайту, та полегшити роботу розробнику.

Перед початком роботи слід встановити необхідні плагіни, встановлення плагінів може відбуватися різними способами, абсолютно такими як завантаження теми на хостинг. Плагіни можуть містити відкритий код, який дає змогу розробникам працювати з плагіном зовсім на іншому рівні, в 90% випадків працювати з плагіном який містить відкритий код куди простіше та краще, ніж з платними плагінами які потребують втручання розробників за деякий кошт.

Розглянемо роботу плагіна на прикладі сайту «Broker» та плагіну «WPBakery Page Builder». Цей плагін не містить відкритого коду, але для роботи з ним це й не потрібно, прикладів подібних плагінів безліч. Продемонструємо завантаження та активування плагіну на рис 4.10.

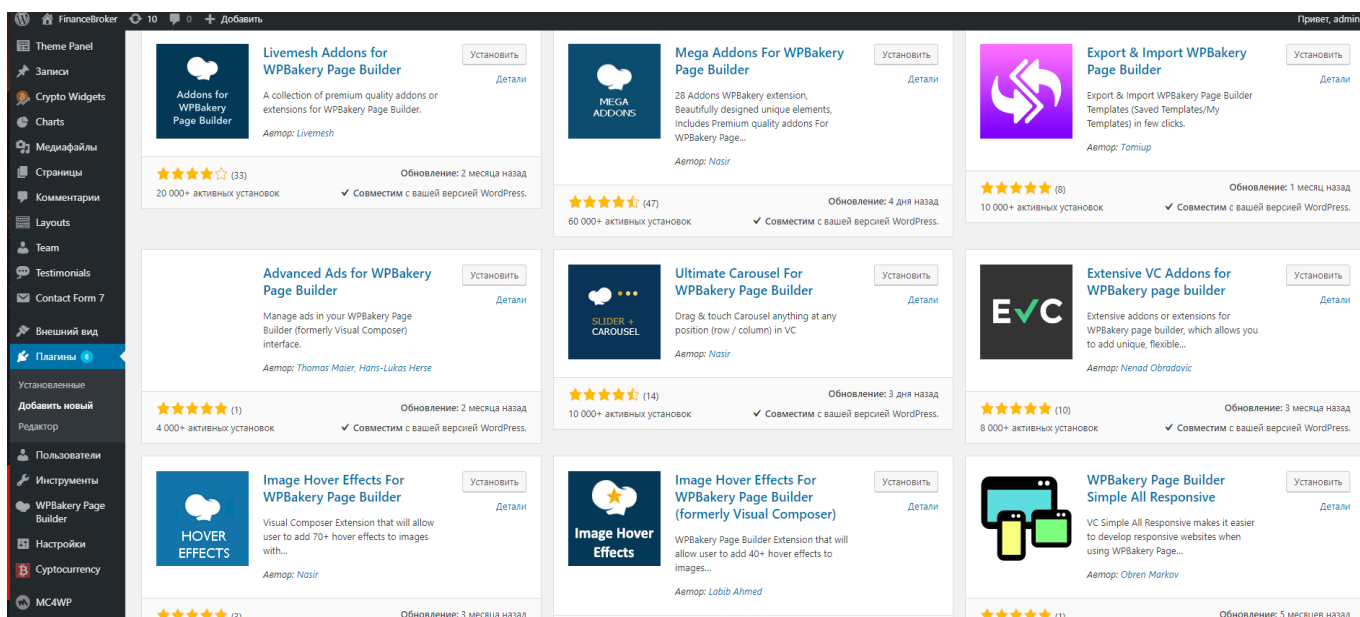


Рис. 4.10 – Встановлення необхідних плагінів

Для вдалого встановлення плагіну, необхідно ретельно ознайомитись з наданими інструкціями, як по шаблону з темою, так і самого плагіна, іноді технічні можливості хостингу просто не дозволять встановити необхідні плагіни, однією з таких проблем, можливо навіть самою популярною є несумісність версії PHP. Після вдалого встановлення плагіну необхідно його активувати, та можна перейти до роботи з ним, в нашому випадку плагін дає змогу будувати певні блоки на сторінці, для розміщення в них інформації, як зображено на рис 4.11.

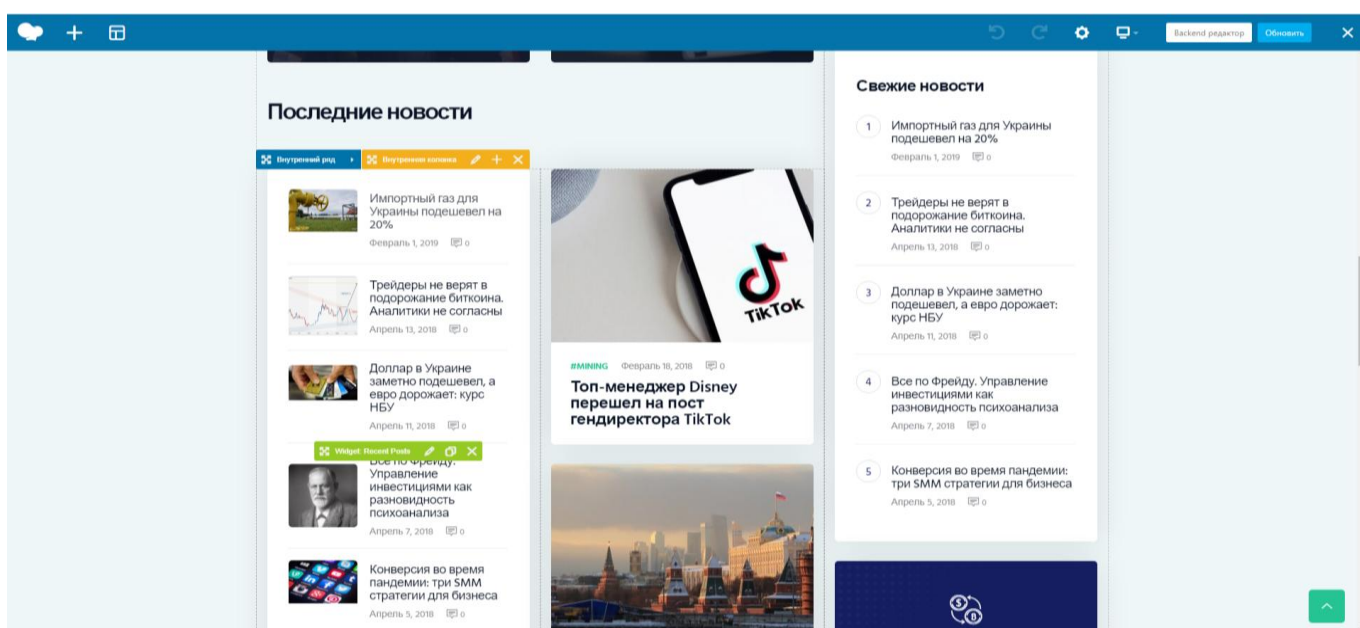


Рис. 4.11 – Робота з PageBuilder

Плагіни подібні до наведеного в прикладі є досить зручними, вони дають змогу працювати як з частиною «front-end» та і «back-end», не потребуючи втручання в CSS, PhP, або інші файли – це дасть змогу власнику сайту за бажанням змінювати розмір та розташування блоків з новинами або будь якою інформацією. Робота плагіну відображена характерним меню блакитного кольору вгорі сайту, а також при наведенні курсором на блок – з’являються більш точні налаштування.

Отже, використання подібних плагінів значно пришвидшує та покращує роботу з сайтом, навіть не знаючи жодної скриптової мови програмування та мов розмітки можна створювати майже шедеври в світі веб-сайтів. Звичайно, ми розуміємо, що плагіни створюють для будь-яких потреб, та для будь-яких сайтів. Ми розглянули необхідний нам плагін, але маємо розуміти, що для коректної роботи шаблону необхідно додатково встановити ще деякі плагіни, наприклад: «Contact Form 7», або «Essential Grid».

4.4. Розробка функціоналу сайту

Сучасні веб-сайти в більшості побудовані за допомогою CMS, це визначає дуже великий рівень попиту на ці послуги та необхідність фахівців в сфері ІТ. Перед розробником була поставлена задача, створити сайт новин за допомогою сучасних технологій та автоматизованих систем, за умовою використання WP, API, JS, CSS, PhP та плагінів з темою, яку передав замовник. Складність та особливість полягає в тому, що більшу частину часу розробник лише налаштовує коректне відображення теми на хостингу, підключає необхідні плагіни та скрипти, а вже потім починається «шліфування» теми.

Під сленговою назвою «шліфування», ми розуміємо необхідність видалення зайвого функціоналу теми, зайвих стилів, скриптів та додавання своїх елементів за допомогою відомих нам та зручних інструментів. Прикладом таких інструментів можуть бути плагіни, які дуже сильно заощаджують час розробника та замовника, або підключення API, який в свою чергу робить ресурс більш оптимізованим та сучасним.

Суть використання API, або готових скриптів з офіційних джерел полягає в тому, що користувач може одразу отримати необхідну інформацію в режимі онлайн, час оновлення інформації може бути досить різним. На прикладі сайту «Broker» буде розглянуто декілька варіантів отримання інформації. Детальніше розглянемо «хедер» сайту, який одразу вітає користувача актуальними котируваннями самих ліквідних валют, рис 4.12.

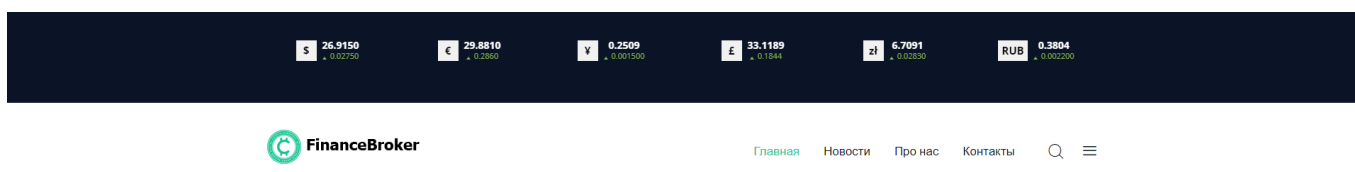


Рис 4.12 – Відображення актуальних котирувань валют

Розробником було вирішено це завдання за допомогою API, від «Приват Банку», яке було встановлено в тему шаблону та задано наслідування класів стилів css для відображення в шапці сайту.

Розглянемо налаштування API в три етапи:

Перший етап – це підключення та налаштування бібліотеки Curl для PHP, бібліотека дає змогу опрацювати отримані данні API, рис 4.13.

```

1 $curl = curl_init($url);
2 if ( !$curl ) die ('Err initial Curl library');
3 // Скачані дані не відображати в потоці
4 curl_setopt($curl,CURLOPT_RETURNTRANSFER,true);
5 // скачування необхідних файлів бібліотеки
6 $page = curl_exec($curl); //вказуємо сторінку де буде відображено роботу бібліотеки
7 curl_close($curl); // Закриваємо з'єднання з сервером бібліотеки
8 // зберігаєм файл

```

Рис 4.13 – Налаштування бібліотеки Curl

Після вдалого налаштування бібліотеки необхідно перейти до другого етапу – етап отримання XML формату від «Приват Банку», бо саме їх API ми будемо застосовувати. Приклад XML для сайту «Broker» зображено на рис 4.14

```

<exchangerates>
  <row>
    <exchangerate ccy="RUR" base_ccy="UAH" buy="0.37817" sale="0.37817"/>
  </row>
  <row>
    <exchangerate ccy="EUR" base_ccy="UAH" buy="18.03042" sale="18.03042"/>
  </row>
  <row>
    <exchangerate ccy="USD" base_ccy="UAH" buy="13.65941" sale="13.65941"/>
  </row>
</exchangerates>

```

Рис 4.15 – XML формат для майбутнього відображення

Наступним кроком буде об'єднати ці скрипти та написати функцію, яка буде обробляти дані курсів, які ми вказали при налаштуванні. По бажанню цю інформацію можна записувати в БД, але треба розуміти, що час оновлення котирувань приблизно 5 хвилин.

Тобто, База Даних через певний проміжок часу буде переповнена інформацією та займатиме багато простору на жорсткому диску, це якраз один з моментів коли за рішенням замовника вирішується чи створювати БД з архівом курсів, чи ні. В нашому випадку, від розробника не потребували збережень інформації, тому розглянемо наступний скрипт, який зображено на рис 4.16.

```

<?php
function _is_curl_installed() {
    if (in_array ('curl', get_loaded_extensions())) {
        return true;
    }
    else {
        return false;
    }
}

function getKurs() {
    global $dna; $dna = true;
    if ( _is_curl_installed() ){
        $url = "https://api.privatbank.ua/p24api/pubinfo?exchange&coursid=5";
        $curl = curl_init($url);
        if ( $curl ){
            curl_setopt($curl,CURLOPT_RETURNTRANSFER,true);
            $page = curl_exec($curl);

            curl_close($curl);
            unset($curl);

            $xml = new SimpleXMLElement($page);
            return $xml->row[2]->exchangerate['sale'][0];
        }
    }
}
?>

```

Рис 4.16 – Поєднання скриптів

Якщо розібрати цей скрипт, то можна зрозуміти, що ми просто поєднали між собою бібліотеку та інформацію з XML, та помістили їх у функцію.

Останнім етапом, буде розглянуто функцію, яка виводить курси валют в «хедер» сайту, скрипт зображено на рис 4.17.

```
$summa = '35$';
$kursUAH = (float)getKurs();
if ($kursUAH > 0){
    // якщо інформація по курсу є (якщо немає то 0)
    echo $summa . ' (int)$summa * $kursUAH ';
} else {
    echo $summa;
}
```

Рис 4.17 – Скрипт виводу інформації

Таким чином, працювати з API лише з першого погляду досить складно, насправді все це лише один із варіантів отримання інформації в режимі реального часу.

Важливо згадати ще один, досить популярний інструмент, який розробник також використав при створенні елементів сайту – це використання «фреймів». Здавалося б, що технологія застаріла, але ще й досі дуже часто ми з вами можемо помічати «інформери» які саме і працюють за допомогою фрейму.

Розглянемо приклад відображення фрейму на рис 4.18.

ИНСТРУМЕНТ	ЦЕНА
BRENT	37.75
WTI	35.41
Газ	1.838
Дата 30.05.2020 19:08 мск	

Рис 4.18 – Відображення роботи фрейму

Цей спосіб значно простіший, навіть початківці зможуть розмістити інформери на свій сайт, суть цього інформеру полягає в тому, що б повідомляти актуальні котирування по маркам сировини нафти та природного газу на російській товарній біржі. Розглянемо код фрейму, який було отримано з офіційно сайту постачальників котирувань, рис 4.19.

```
69 <iframe style="width:100%;border:0;overflow:hidden;background-color:transparent;height:314px"
70 scrolling="no"
71 src="https://fortrader.org/informers/getInformer?st=1&cat=13&title=(назва ресурсів)texts=(валюти)mult=0.95&showGetBtn=1&w=0&colors=false&items=(назва колонок)=bid"></iframe>
```

Рис 4.19 – Відображення інформеру

Для розміщення коду, який зображений на рис.4.19, було видалено зашифровану інформацію в вигляді дуже великої кількості символів та замінено на надписи в дужках. Звичайно, що цей скрипт не буде працювати, але можна подивитись саму суть роботи фрейму.

Ще одним можливим варіантом отримання актуальних даних, є розміщення власних ручних графіків котирувань. Справа в тому, що наразі існує безліч сайтів в яких інформацію вводять руками, а не отримують з джерел, звичайно що ця процедура застаріла, але іноді вона необхідна. Розглянемо відображення створеного графіку без API, рис 4.20.



Рис 4.20 – Відображення графіку створеного руками

Насправді, цей графік можна створити за допомогою деякого плагіну, який дає змогу зчитувати таблиці з EXEL, та формувати їх в такий вигляд, тому продемонструємо роботу цього плагіну та його налаштування на рис 4.21.

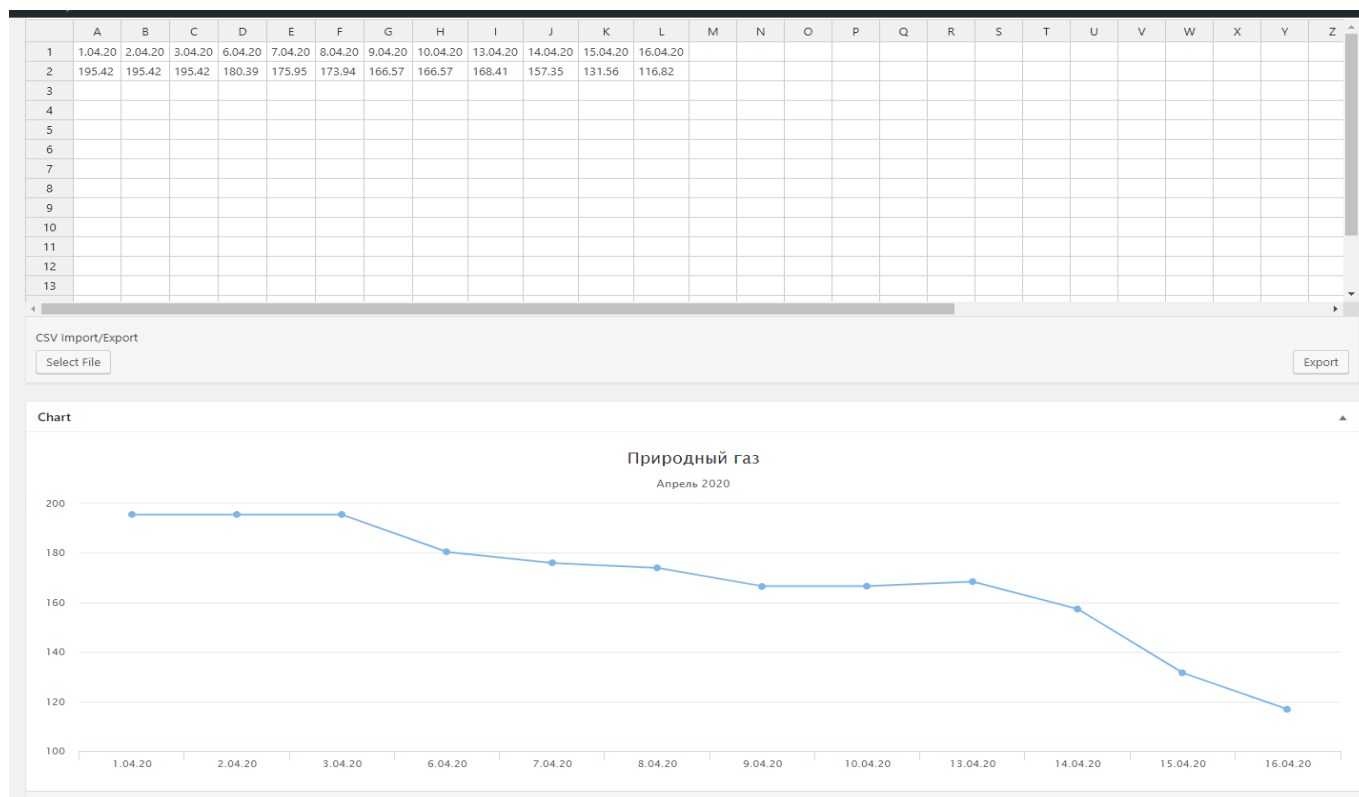


Рис 4.21 – Відображення роботи плагіну «Crypto Widgets».

Зверху можна побачити, що відображена звичайна таблиця як в Excel. Перший рядок – це виставлення дат, другий рядок – це виставлення цін на дату, такий вид графіку розрахований на оновлення інформації кожні 24 години, редакцією сайту, що може нести в собі певні незручності, але і має безліч позитивних сторін.

У користувача є можливість зберігати таблиці з графіками у форматі .xlsx, а також повторне завантаження на сайт та зберігання в БД.

Розглянемо ще одну унікальну функцію на сайті – це графік котирувань в реальному часі, згідно торгових сесій на світовій біржі NASDAQ, яку користувачі можуть побачити лише використовуючи комп'ютер, на жаль з телефону відобразити таку інформацію не можна, як зображено на рис 4.22.



Рис 4.22 – Відображення графіку котирувань онлайн

Звичайно, що це не весь функціонал сайту, але це самі головні моменти back-end розробки яка була проведена. Сайт містить окремі блоки новин, блоки коментарів та окремі сторінки про майбутню команду редакторів сайту, а також контактну форму створену за допомогою популярного плагіну.

Звернемо увагу, на стилістику сайту та яскраві моменти дизайну, та розглянемо на прикладі сторінки «Про Нас», зображену на рис 4.23.

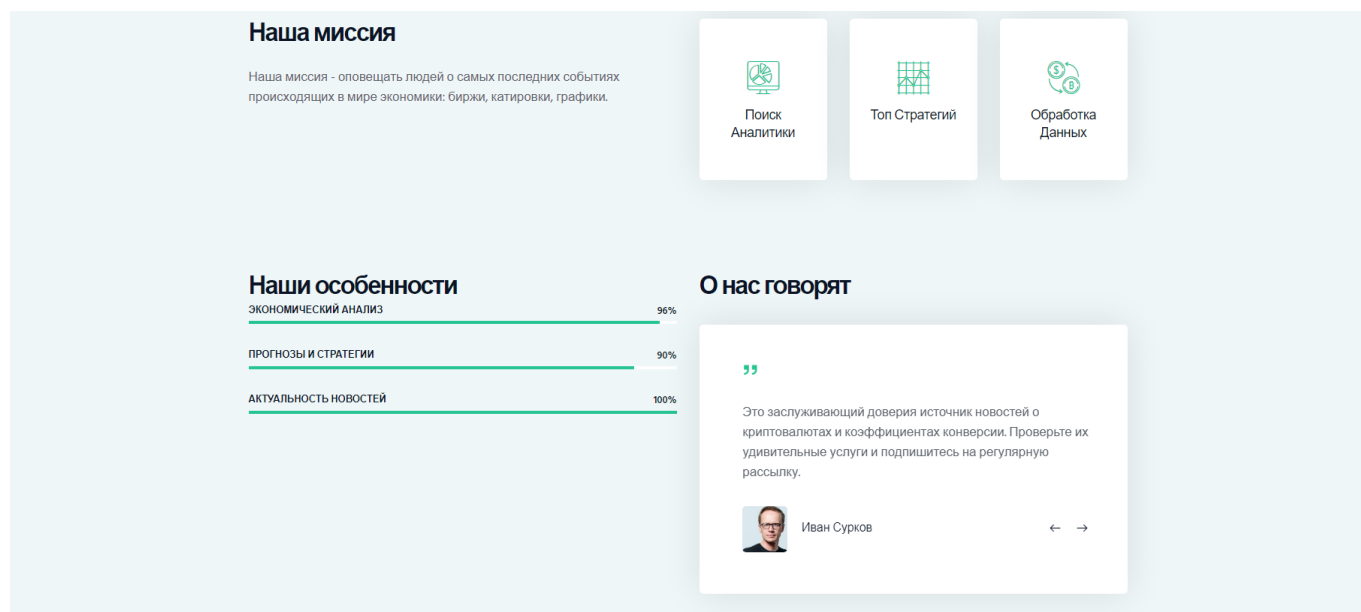


Рис 4.23– Дизайн сторінки «Про Нас»

Сторінка містить в собі елементи слайдеру «карусель», що зроблена за допомогою відкритих скриптів JQuery, а також гарний CSS елемент з шкалою особливостей.

Підсумком роботи – демонстрація головної сторінки ресурсу з частиною елементів розміщених на ній, при прокрутці сторінки будуть з’являтися нові блоки новин, та слайдер в хедері сайту, рис 4.24.

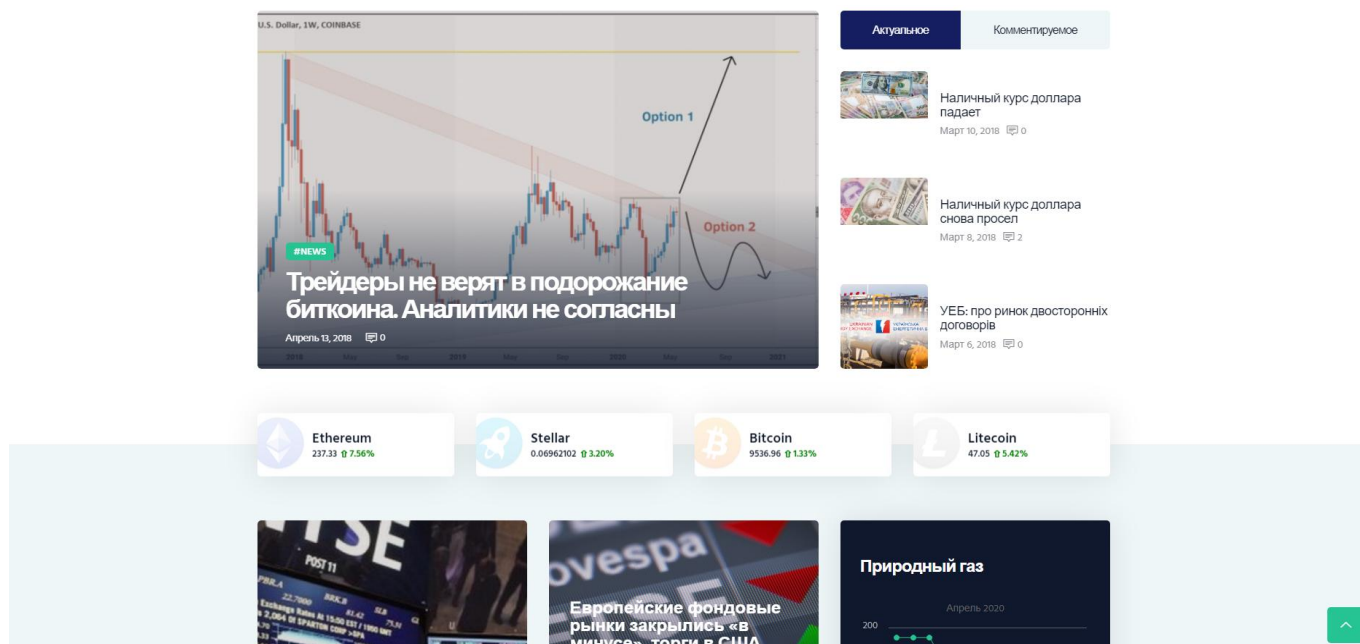


Рис 4.24 – Загальний вигляд головної сторінки

Отже, розробником, було виконано поставлені завдання від замовника, за допомогою сучасних інструментів розробки та використання CMS WP. Особливими інструментами розробки стали API, які дають змогу отримувати бажану інформацію в режимі реального часу від офіційних постачальників котирувань. З адміністративної частини сайту, для власника є змога створювати динамічні новини та розміщувати їх згідно категорій на сторінках сайту, без обмежень. Всі ці новини можуть зберігатись в БД та в майбутньому може бути створений архів новин, якщо це буде необхідно.

Кращим додатком до сайту, є підключення скрипту, який дає змогу «парсити» новини з других джерел, тобто повністю автоматизувати розміщення новин без

допомоги редактора, дуже багато сучасних українських інтернет-видань використовують ці скрипти.

4.5. Тестування сайту

Тестування – це оцінка якості веб-сайту, методом експериментальної перевірки, та шляхом виконання тестів. Мета тестування – виявити наявність помилок, неузгодженостей, бекдорів.

Кінцевою метою тестування – є повноцінна та якісна оцінка веб-ресурсу зі сторони замовника.

У випадку сайту «Broker», сайт було протестовано, як замовником, так і розробником на виконання алгоритму сортування, та на коректне відображення на різних пристроях, за деякими виключеннями. Продемонструємо відображення сайту на мобільних пристроях, на прикладі Iphone X, як зображено на рис. 4.25.

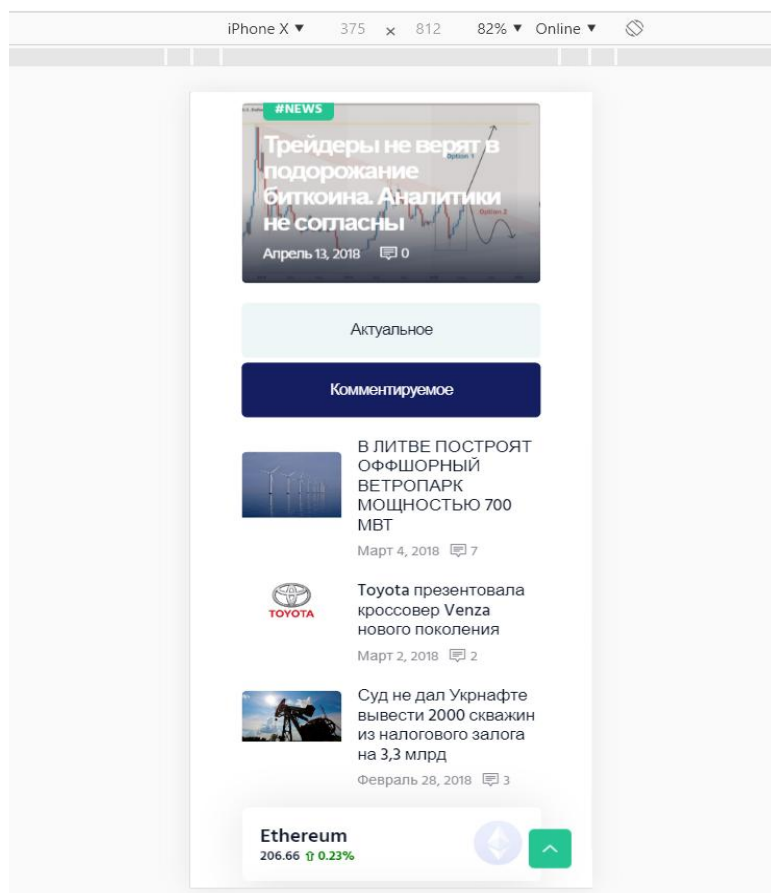


Рис. 4.25 – Відображення на мобільному пристрої

Загалом, саме питання тестування – це досить відносне питання, на яке важко відповісти розробникам, адже не існує чітких норм та стандартів які стосуються веб-сайті. Ще декілька років тому тестування сайту взагалі не розглядалось як необхідність, а в сучасному ІТ бізнесі існують навіть окремі ланки тестувальників, котрі називаються QA engineering.

Додатково розглянемо відображення на найменш популярному інтернет-браузері «Explorer» від компанії Майкрософт, як зображено на рис 4.26.

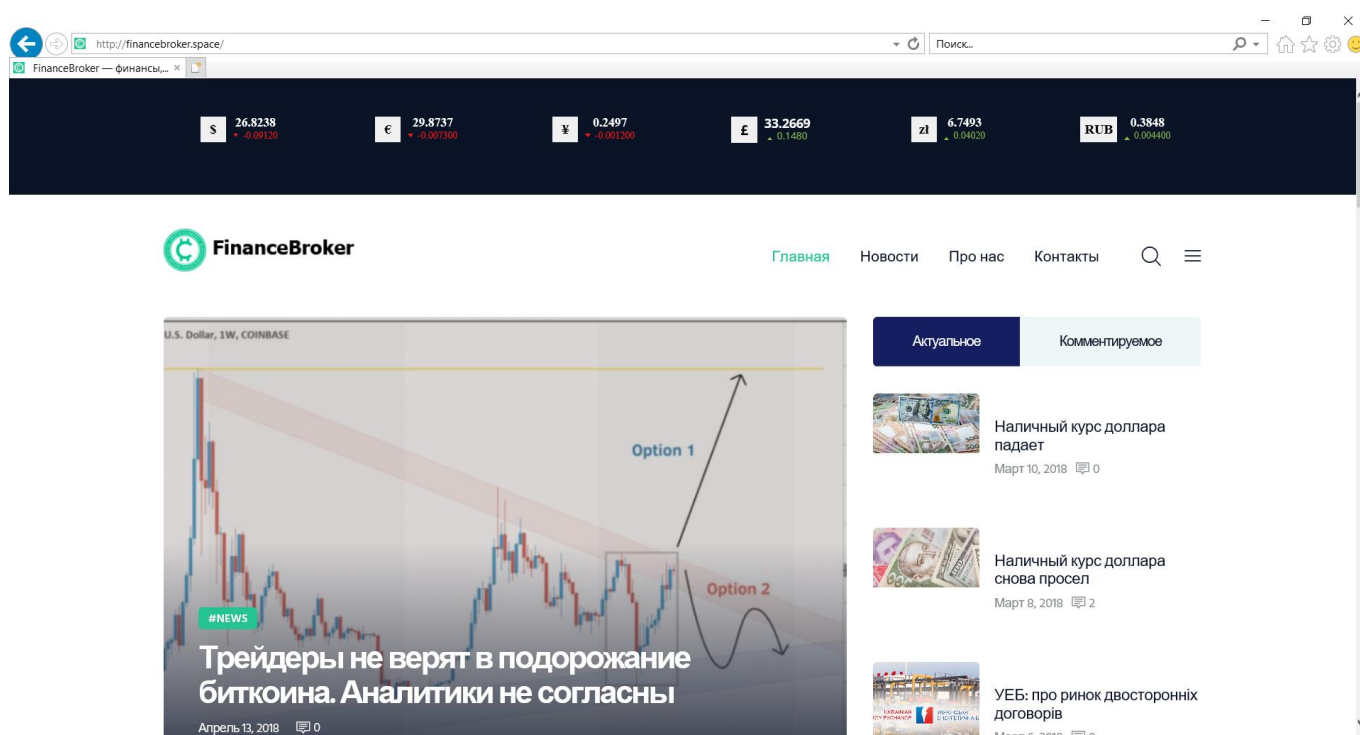


Рис 4.26 – Відображення в браузері «Explorer»

Як ми можемо зрозуміти, ніяких проблем в відображенні інформації на сайті не знайдено, тому завдання виконане вірно.

ВИСНОВОК

За результатами даного дипломного проекту, був розроблений сайт фінансових новин «Broker». При створенні сайту, були використані сучасні технології, у вигляді API від різних компаній-постачальників котирувань. Розроблена фізична та логічна структура сайту в більш спрощеному вигляді. Дотримано всі вимоги технічного завдання та проведено тестування на різних браузерях та мобільних пристроях.

Всі розробки велись за допомогою скриптових мов програмування, таких як: PHP, Js , Ajax, та різних плагінів.

Самостійно було розроблено основні елементами сайту, а саме головна сторінка сайту, форма зв'язку з адміністраторами сайту та розміщення віджетів, для відображення котирувань, та курсів валют, за допомогою API, «фреймів», а також плагіну «Crypto Widgets».

Головними порадами в подальшій розробці та використанні сайту, є підключення SSL-сертифікату, який шифрує підключення користувачів та дає змогу проводити навіть банківські операції з захистом даних, та приховувати паролі користувачів. А також, оптимізація роботи PHP скриптів, у вигляді видалення зайвих фрагментів коду від функціоналу, який зовсім не потрібний.

Створений сайт фінансових новин «Broker» розташований в мережі Internet, і прийнятий замовником, про що свідчить акт впровадження.

СПИСОК ЛІТЕРАТУРИ

1. Берковський В.В., Радіонов В.О., Бурковський В.С. Зіставлення технологій створення сайтів з їх цільовим призначенням / В.В. Берковський, В.О. Радіонов, В.С. Бурковський // Наука і техніка Повітряних Сил Збройних Сил України. – 2015. – № 2(19). – С. 122-124.
2. Вовк О. Б. Методи та засоби підвищення життєздатності веб-сайту як інформаційного продукту: автореф. дис. ... канд. техн. наук: 05.13.06 / Вовк Олена Борисівна; М-во освіти і науки України, Нац. ун-т «Львів. Політехніка». – Львів, 2013. – 20 с.: іл., табл. – Бібліогр.: с. 17-18.
3. Горностаева Е. А. Современные проблемы дизайна сайтов и вызовы нового времени / Горностаева Е. А. // Молодой ученый. — 2015. — №1. — С. 38-40.
4. Дакетт Д. JavaScript и jQuery. Интерактивная веб-разработка / Д. Дакетт. – Москва: Издательство „Э”, 2017. – 640 с.
5. Дорофеева О. Удосконалення доступу до галузевого веб-ресурсу / Ольга Дорофеева // Бібліотечний вісник. – 2010. – № 1. – С. 3-9.
6. Евдокимов Н.С. Создание сайтов / Н.С. Евдокимов. – Санкт-Петербург: Питер, 2014. – 410 с.
7. Інькова Н. А. Створення Web-сайтів : Навчально-методичний посібник (Електронний ресурс) / Інькова Н. А., Зайцева Е. А., Кузьміна Н. В., Толстих С. Г. – Режим доступу до електронного ресурсу: <http://club-edu.tambov.ru/methodic/fio/p5.doc> (дата звернення 29.05.2020).
8. Компанеєтс М.О. Принципи проектування ефективних вебсайтів / М.О. Компанеєтс // Молодий вчений. – 2015. – № 9. – Част. 2. – С. 106-108.
9. Котеров Д.В. PHP 7 / Д.В. Котеров. – Санкт-Петербург: БХВ-Петербург, 2016. – 1088 с. Колисниченко Д.Н. PHP и SQL. Разработка Web-приложений. / Д.Н. Колисниченко. – СПб.: БХВ-Петербург, 2013. – 543 с.
10. Матат Д. Нові технології: веб-квест / Дарина Матат // Освіта України. – 2014. – 23 черв. (№ 23). – С. 10.

11. Мороз Виктор. Защита авторских прав на веб-сайт / Виктор Мороз // Юридическая Практика. – 2010. – № 16(643). – С. 18-19.
12. Основні етапи розробки веб-сайту (Електронний ресурс) / [укл. Юрчак І. Ю.] / Національний університет "Львівська політехніка". – Режим доступу до електронного ресурсу : <http://www.victoria.lviv.ua/html/wp/sssteps.html> (дата звернення 29.05.2020).
13. Паньчик Б. Розробка ефективних веб-додатків (текст на англ. мові) / Беата Паньчик // Актуальні проблеми економіки. – 2013. – № 11. – С. 443-251.
14. Преимущества web-сайтов на CMS (Електронний ресурс). Режим доступу до електронного ресурсу – <http://webfactura.ru/services/website/advantage/> (дата звернення 29.05.2020).
15. Система управления сайтом (CMS) (Електронний ресурс). Режим доступу до електронного ресурсу – <http://www.grizliart.ru/cms/> (дата звернення 29.05.2020).
16. Тарнавська Т.В. Сутність інформаційних технологій в освіті / Т.В. Тарнавська // – К.: Кондор. – 2013. – 210с
17. Угрин Д. І. Веб-технології та веб-дизайн: [навч. посіб. для студ. вищ. навч. закл.] / Д. І. Угрин, М. І. Маниліч, Б. Є. Деркач. – Чернівці : Золоті литаври, 2012. – 285 с.: іл., табл. – Бібліогр.: с. 284-285.
18. Філінович В. В. Веб-сайт як особливий об'єкт права інтелектуальної власності: автореф. дис. ... канд. юрид. наук: 12.00.03 / Філінович Валерія Вікторівна; М-во освіти і науки України, Київ. нац. ун-т ім. Т. Шевченка. – Київ, 2015. – 19 с. – Бібліогр.: с. 15-17.

ДОДАТОК А.**Програма**